Predicting with Structured Data:
Graphs, Ranks, and Time Series

構造化データに対する予測手法：グラフ、順序、時系列

Jiuding Duan
段 九鼎

ABSTRACT

Predictive models have received wide attractions in modern engineering, financial, and social problems. We develop models and algorithms to analyse the patterns in big data, and try to leverage these patterns to make better predictions on what may happen in the next. In this process, structured data and patterns play an important role, because they are the intersections between the raw observations that we collect from the real world and the algorithms that we build and operate on computing resources. In the pursuit of favourable predictive performance, the models are mainly challenged by the variety and veracity of structured data and representations, because we neither have trustworthy raw observations that unveil the ground truth nor have enough amount of qualitative techniques to infer the values of interest given the heterogeneous observations.

In this dissertation, we attempt to address the variety and veracity problem in predicting with structured data by developing learning algorithms that can discover hidden structured patterns and use the patterns to facilitate the learning algorithm and the prediction. There are three essential challenges involved: (1) develop a method for the graph-structured data in materials informatics so that the fine-grained similarity of the graph elements can be accounted and contributes to the prediction, (2) construct learning algorithms and proper data representations so that the risk of conflicting observations can be mitigated, (3) develop time series forecasting algorithm that is robust to noisy temporal observation and is complementary to legacy evaluation metrics.

To tackle these challenges, we propose innovative structured learning techniques that can incorporate structured patterns as complementary input, output, and model components that can be learned directly from raw observations. For the first challenge, we develop fine-

grained kernels to describe the similarity of graph elements, e.g., labeled vertices and labeled edges. The proposed technique is tailored for domain experts to contribute and transmit their knowledge to the kernel construction. The proposed method achieves favourable predictive performance compared to the methods that use handmade features. In the small data scenario in materials informatics, the proposed method exhibits a significant advantage when the availability of annotations is limited. For the second challenge, we propose two techniques ranging from shallow model to deep neural networks to handle the intransitive relationships in raw observations. The models generalise the related works via incorporating cross-sectional numerical interactions between model parameters. For the shallow model, we define the numerical interactions via additional matrices and show constructively how the model generalise to the legacy related works. For the deep neural networks, we devise a structured neural network that simplifies the legacy models while maintaining the notion of pairwise comparison and matchup. A thorough investigation on real-world datasets show an universal presence of conflicting observations and highlights the importance of developing algorithms that can handle intransitive relationships. For the final challenge, we develop an alternative perspective, i.e., the ranking perspective, that is robust to noisy temporal observations in comparison with the legacy methods that optimize tracking errors. The proposed forecasting algorithm leverages the learning-to-rank technique and can simultaneously analyze the time series from both ranking and tracking error perspective by adopting a local learning algorithm that enables augmented inference. We are able to obtain improved ranking performance for the temporal observations, and moreover, this improvement is complementary to the tracking errors that are optimized by the conventional methods.

Acknowledgements

First, I would like to express my great appreciation to all the people I met during my doctoral study. I would like to thank my advisor, Prof. Hisashi Kashima. He kindly gave me tremendous support and guidance in the fantastic world of machine learning and data mining research. His motivating ideas and advises were invaluable to me, as well as to what I have achieved during my doctoral study. I would also like to thank Prof. Akihiro Yamamoto and Prof. Tatsuya Akutsu who were the committee members of my thesis and reviewed the thesis with profound insights that help improve the thesis and the presentation.

I would also like to thank my collaborators who kindly shared their research expertise and experience in my research works. I would like to express gratitude to Prof. Atsuto Seko for introducing me to the area of materials informatics and instructing the terminologies and the related problems in the area. I would also like to thank Dr. Shotaro Miwa, Prof. Hiroto Saigo, and Makoto Kushino for offering me prestigious R&D opportunities to work on exciting problems in various domains. I would like to thank Prof. Yukino Baba and Prof. Marco Cuturi for organizing seminar events and projects in and out of the education curriculum. A special thank goes to Naoki Otani, Dr. Jiyi Li, Dr. Shogo Hayashi, Dr. Akifumi Okuno, Yanghua Jin, Guoxi Zhang, Zebang Chen, Dr. Yanchun Jin, Dr. Yan Gu, Dr. Ruining Yang, Dr. Xueni Gu, and many others who kindly shared their enjoyable life experience with me in and out of the lab.

Finally, I would like to express my appreciation to my family members for their continuous encouragement and decades of support, without which all my achievements would not be possible.

Contents

List of Figures

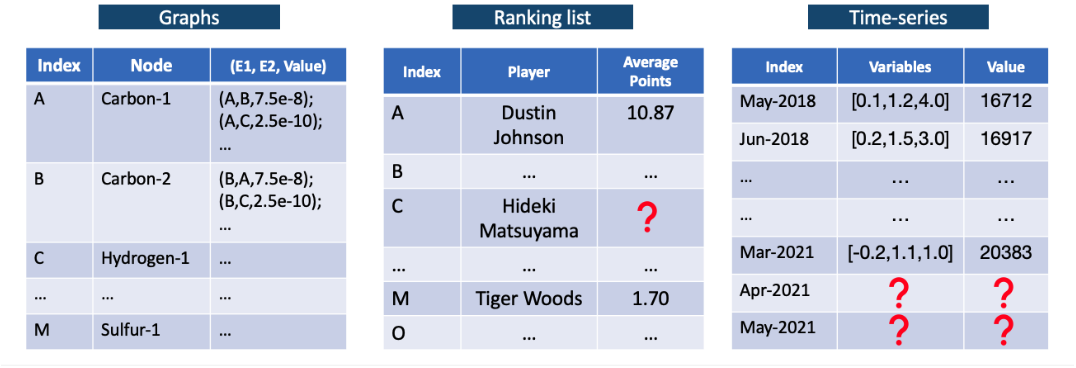
List of Tables

Chapter 1

Introduction

When we encounter a predictive analytics task for big data, existing solutions or algorithms are often challenged by the 4Vs, i.e., volume, velocity, variety and veracity of the data. To deal with these challenges, one may design elegant algorithms and data-driven models by leveraging the structure of the data in efficient and effective ways; for example, applying the infamous "kernel trick" on high-dimensional data helps transform the numerical optimization of an empirical risk minimization into solving for a well-posed linear system. Others may tailor the structure and units of deep neural networks depending on how it would best fit the characteristics of data [1]. An example of such is the adversarial training techniques that are developed to mitigate the risk of over-fitting when random or even adversarial samples are involved in model training. Therefore, it is well recognized that a proper structuring of data representation and additional efforts on feature engineering can provide valuable reflections on how to enhance the performance of a predictive model, despite that certain downstream problems like finding a global optimum and maintaining the generalization power of a model are equally important in the implementation.

A proper structuring of data for machine learning responses to the 4V challenges in different ways. First, on volume, machine learning has been developed with many theoretical justifications, generic techniques to create learning algorithms with structured data remain



Graphs

Index	Node	(E1, E2, Value)

A	Carbon-1	(A,B,7.5e-8);
		(A,C,2.5e-10);
		...

B	Carbon-2	(B,A,7.5e-8);
		(B,C,2.5e-10);
		...

C	Hydrogen-1	...
...	...	...
M	Sulfur-1	...

Ranking list

Index	Player	Average Points
A	Dustin Johnson	10.87
B	...	...
C	Hideki Matsuyama	?
...	...	...
M	Tiger Woods	1.70
O	...	...

Time-series

Index	Variables	Value
May-2018	[0.1,1.2,4.0]	16712
Jun-2018	[0.2,1.5,3.0]	16917
...	...	...
...	...	...
Mar-2021	[-0.2,1.1,1.0]	20383
Apr-2021	?	?
May-2021	?	?

Figure 1.1: Data representations for graphs, ranking lists and time series.

as an open area when the algorithms that operate on structured data are analysed in terms of complexity. Second, on velocity, when data structure is weaved into the data management process, the technical challenge tends to shift it focus away from machine learning to the acceleration of numerical computation and the deployment of data operations. Third, on variety, the overwhelming presence of unstructured data, such as images and audios that do not easily fit into a tabular representation on a spreadsheet or in a relational database, enlarges the information gap between raw data and extracted patterns. Lastly, on veracity, in the process of translating a problem from its primitive description to a solvable mathematical formulation, some meaningful information may be lost and the intrusion of toxic information may be unavoidable, so a proper representation of data is needed to carry on the profound insights. Overall, all the 4V challenges in big data are exposed to the problem of structuring data in different ways and machine learning is a generic and promising solution to these challenges.

Among the multi-facet deliberations of data structure and algorithms, leveraging the structured patterns hidden behind raw observations is an all-weather technique to enhance the utility of a machine learning model or a predictive modeling workflow. For example, in materials informatics, the domain knowledge of similarities between atomic elements can be critical to the accurate inference of certain material properties; fine-grained atom-level information is descriptive of electromagnetic properties, because the formation process of the material compound is essentially a series of electromagnetic interactions between the underlying atoms.

In network analysis and preference aggregation problem, feature extraction is difficult from raw observation, so this challenge on how to structure effective features from the nodes and objects motivates the development of representation learning techniques, which aim to build learning algorithms that can automatically discover favorable hidden patterns from the data. In Fig. 1.1, three typical data representations, i.e., graphs in materials informatics, ranking list in recommender system, and time-series in the financial domain are illustrated to highlight the variation of problem settings in predictive modeling, especially when the variety and veracity issue of big data is concerned.

In recent years, a transition from statistical machine learning models to deep neural networks, which aim to integrate feature extraction into learning algorithms in an end-to-end manner, has been witnessed in the research community. In this transition, the gap between the number of available techniques to discover hidden patterns and the exponential accumulation of unstructured data is large. As a result of the gap, feature engineering is no longer a standalone module in a predictive modeling workflow. Instead, more and more features can be learned directly from raw data via representation learning algorithms [5]. For example, word embedding techniques in natural language processing (NLP) has been widely adopted to improve the performance over handicraft NLP features. By learning a multi-dimensional representation for the word QUEEN from the data, the learned word representations have demonstrated superior descriptive power as a feature vector and have exhibited favourable cognitive properties in natural language, which can be as simple as the following semantic "equation":

$$\overbrace{[2.0, 1.0, 4.1]}^{\text{QUEEN}} \Leftarrow \overbrace{[0.9, 1.1, 2.0]}^{\text{KING}} - \overbrace{[-1, 0.5, -2.0]}^{\text{MAN}} + \overbrace{[0.1, 0.5, 4.0]}^{\text{WOMAN}}$$

Overall, it has been recognized that revisiting the data representation and its utility is a prerequisite in the successful design of learning algorithms that are challenged by the 4Vs in big data. In order to better discover and leverage the hidden patterns from noisy observations and limited source of annotations, several directions are envisioned by the research community.

The most intuitive direction is to restructure the input and ensure that meaningful information are fed into the model. A similar mindset can be applied to the output of models to ensure meaningful feedback signals can be back-propagated to facilitate the parameter estimation. In statistical machine learning, the iterative learning and inference procedure provides a fertile soil to propagate structured patterns back and force; therefore, revisiting the algorithmic challenges from alternative angles and resolutions is thrilling, albeit with some technical difficulties.

1.1 Challenges

In practice, the elements of structured data are sometimes only partially observable. This happens either due to a high measurement or assessment cost to obtain an accurate and trustworthy observation or because the ill-posed problem definition inevitably involves conflicting observations. For example, the functional property of a material compound is unveiled to material scientists before the material is physically developed with a high implementation cost. As a result, in order to guide the high-throughput materials discovery and better control the cost, predictive models are positioned to leverage only a limited amount of graph-structured observations, although the patterns and annotations are domain-specific and difficult to handle by generic models. To overcome this shortage of annotations, learning algorithms that can outperform with small data are demanded. Indeed, such a shortage of annotated data is common to the data science problems in many scientific discovery [2], not only in materials informatics but also in the development of machine learning algorithms in pharmaceutical industry, medical diagnosis, earth observation and astronomy. Besides, the technical challenge caused by conflicting or adversarial observations is also inevitable. In rank aggregation, in order to predict a result exhibiting a cross-sectional and temporal consistency, the learning algorithm has to arbitrarily eliminate conflicting observations. This is only feasible with an awareness of hidden patterns behind the model. In this dissertation, several approaches are proposed to tackle the problem of predicting with structured data with a focus on the variety and veracity of structured data. By taking one step back from the conventional problem settings, the proposed approaches first revisit the hidden pattern behind the raw observations and then deliberately solve the problems by raising the following fundamental questions:

1. **DOES** atomic-level similarity provide constructive information for predictive analytics in material informatics?

2. **WHAT** is the proper data representation to mitigate the risk of conflicting observations, given unconcordant pairwise relationships? **HOW** to improve the predictive performance?

3. **IS** concordance an alternative and meaningful optimization objective for time-series forecasting? **HOW** to utilize the concordance information as a complement to the conventional approaches that optimize tracking errors?

For Question 1, we face a scarcity of effective features related to the data variety. A conventional and *de facto* approach in statistical machine learning to deal with such a scarcity of effective features and annotated data is kernel method. The kernel method constructs kernel matrices by exploiting the similarity of objects, and thereafter learns a predictive model from the data. Although the kernelization of machine learning models for a variety of structured data has been extensively studied in literature, the techniques regarding how to utilize domain knowledge in random walk graph kernels for materials informatics problem remain unexplored for both material scientists and computer scientists.

In Chapter 2, we propose a novel predictive model for materials informatics. The proposed model incorporates fine-grained details, e.g., atomic-level and bond-level similarities of the graph elements in materials compounds by using graph kernel. To calculate the similarity at atom level and bond level, we refer to one of the most fundamental domain knowledge in materials science and chemistry, i.e., the periodic table of the elements. In the periodic table, the element-level adjacency in column and in row are qualitative descriptors for electromagnetic properties. Thus, we can use the concordance relationship between the random walk sequences

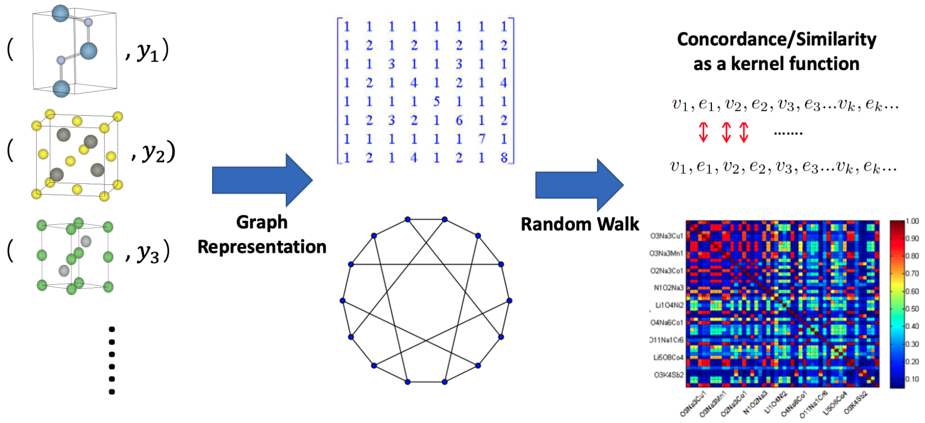


Figure 1.2: Predictive modeling pipeline for graph-structured data in materials informatics.

on graphs as a proxy for the estimation of the similarity between material compounds by adopting soft matching functions instead of the exact matching functions in the primitive definition. To reduce the complexity of the computation, we formulate a minimum spanning tree algorithm that reduces the complexity of of a kernel update from $O(|V|^3)$ to $O(|V|^2)$ by cutting off the redundant chemical bonds in a graph representation. The proposed method is examined in real world materials informatics database. In comparison to the methods that rely on conventional handicraft features, the proposed method exhibited superior predictive performance when only a tiny fraction of labeled samples is available for training. This is favorable for the small data challenge in materials informatics.

For Question 2, we study the modeling of intransitive relationships. Intransitive relationship refers to a property of binary relations that are not transitive. The problem is a derivative of data veracity and is unique to graph and ranking data. In network and ranking analysis, one may gather ranking information from different data vendors or annotators, while the annotators may provide data with limited veracity intentionally or unintentionally. It is trivial to show that even for the simplest form of ranking observation, i,e., pairwise comparison and matchup, the intransitivity issue is unavoidable if the pairwise relationships can propagate freely on different ranking lists. To infer an unobserved pairwise relationship, one can distill the conflicting observations by arbitrarily propagating the relationships and dropping out the

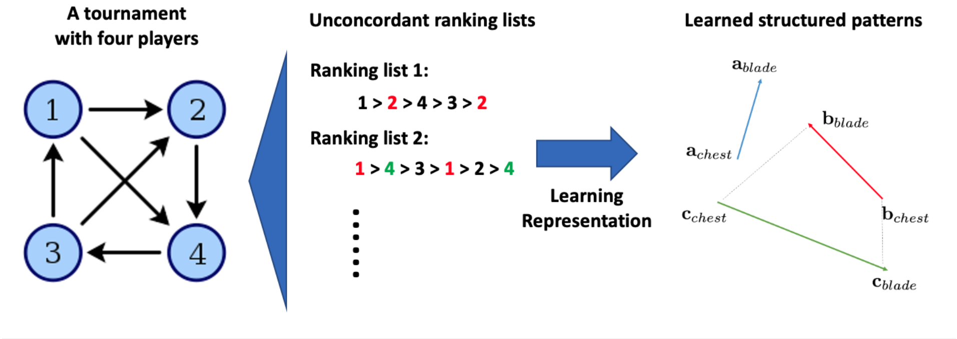
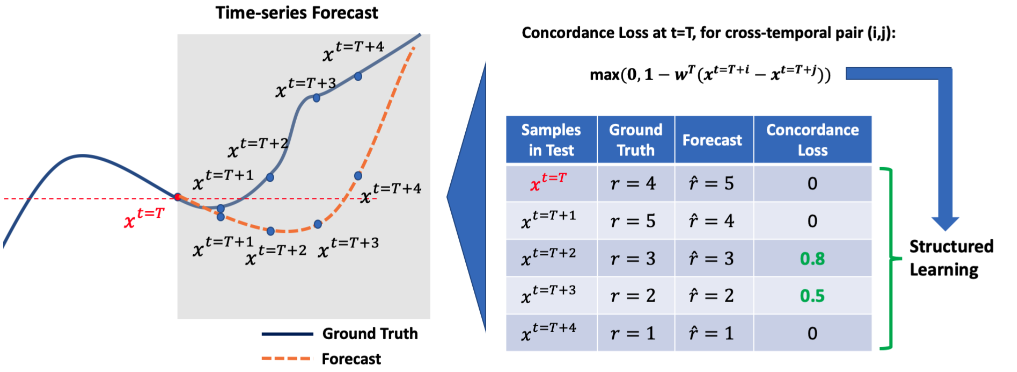


Figure 1.3: Learning structured representation of objects for predictive modeling of intransitive pairwise relationships.

conflicts along the path. However, it is not always preferred to do so, because the uncertainty of pairwise relationships is nature to the problem and such uncertainty has to hold a presence in the predictive model. In order to systematically handle the conflicting observations, a proper structuring of data representation is needed.

In Chapter 3, we present a novel multi-dimensional representation learning technique for predicting pairwise comparison results. Our proposed method utilizes a unified structured representation of objects that can represent various scoring attributions, including the interaction between objects and the intrinsic strength of each object. On the expressiveness of the proposed model, we provide a simple and constructive example that shows how the proposed model generalises its related works. For the estimation of the parameters, we devise a stochastic gradient algorithm based on alternating direction methods of multipliers. Through missing value inference experiments on real-world datasets covering recommender system, food preference and online gaming platform, we demonstrate that the proposed method improves the prediction accuracy of existing models. Moreover, to highlight the importance of dealing with intransitivity in data science problems that involve a ranking perspective, we provide an extensive quantitative investigation on the universal existence of intransitive relationships in real-world datasets.

In Chapter 4, we further present an end-to-end intransitivity modeling technique for learning with pairwise comparisons and matchups. Different from the techniques presented in Chapter



Samples in Test	Ground Truth	Forecast	Concordance Loss
$x^{t=T}$	$r = 4$	$\hat{r} = 5$	0
$x^{t=T+1}$	$r = 5$	$\hat{r} = 4$	0
$x^{t=T+2}$	$r = 3$	$\hat{r} = 3$	0.8
$x^{t=T+3}$	$r = 2$	$\hat{r} = 2$	0.5
$x^{t=T+4}$	$r = 1$	$\hat{r} = 1$	0

Figure 1.4: Learning to rank temporal observation pairs for time-series forecasting.

3, a structured multi-layer neural network is used for the parameter estimation, instead of the conventional logistic function. Through experiments on synthetic data with a controlled rank, we verified that the proposed end-to-end approach is superior to shallow models that do not leverage deep neural networks. In comparison to the model presented in Chapter 3, the proposed technique opens the gateway to future research and development of more expressive network structures. Through missing value inference experiments, we demonstrate that the proposed method is a simpler yet more effective method to improve the prediction accuracy for the modeling of pairwise comparison and matchup.

For Question 3, we concern the veracity of temporal data. Time-series forecasting is a common research topic in this scope. Forecasting requires a model capacity to capture the long-term dependency in the past and to make prediction of future events in a forward-looking horizon. Conventional evaluation metrics of time-series forecasting are unthinkingly based on tracking error and are not robust to inaccurate measurements of temporal observations. When the utility of the forecasts is evaluated in the lens of relative orderings of the temporal observations, tracking error can be too primitive because it does not help minimize the misalignment between the forecast and the ground truth. In the financial domain, asset managers and hedge funds may prefer a longer forward-looking horizon and weights the utility of relative ordering between temporal observations more than that of the tracking errors. In such cases, learning algorithms that can optimize the ranking objectives directly is appreciated, because it maximises the utility of the model output and is more robust considering discrete nature of the model outputs.

In Chapter 5, we propose a novel time-series forecasting method that optimizes and infers the relative relationship of temporal observations by using learning to rank technique. A local learning technique for temporal observations based on nearest neighbor model is proposed to make the discrete forecast comparable to what is predicted by the conventional tracking-error-focused models. To improve the predictive performance, heterogeneous modules are wrapped up in a dynamic prediction strategy. Extensive experiments on real-world financial data show that the proposed method significantly improves the prediction of relative relationship between temporal observations. Moreover, the utility of the model is discussed for different types of time-series and within various lengths of forward-looking horizon.

Chapter 2

Predictive Modeling for Graph-structured Data in Materials Informatics

2.1 Introduction

Machine learning with structured data has been widely applied to computer vision, natural language processing, speech recognition, bioinformatics, and etc. In order to achieve the goal of certain domain-specific predictive or prescriptive analytical tasks with satisfactory, researchers in informatics as well as experts in the respective domain investigate various scientific data and patterns. The inspirations come in two perspectives. The first perspective is the algorithmic perspective that focuses on data representations and efficient algorithms that manipulate underlying data structures. The second perspective is the domain perspective, that focuses on the discovery of patterns aiming at a better construction of logical explanations based on domain knowledge. Both perspectives are indispensable for a successful design of data-driven algorithm based on machine learning and data mining techniques [3].

Recently, the rapid growth of volume and velocity of big data has made impacts on scientific discovery. Materials informatics [4], a subarea in scientific discovery, has been pioneered by the Materials Genomes Initiative [35] and thereafter followed by researchers who develop novel algorithms that can combine the machine learning algorithms with interdisciplinary knowledge or insights from domain experts in chemistry and physics. Examples of such advancement include the prediction of material properties that are difficult to measure or simulate and the discovery of novel functional materials based on a limited amount of experimental trails. Over time, more and more successful applications have emerged, not only in materials informatics, but also in the pharmaceutical industry, in earth observation and in astronomy.

In materials informatics, material compounds are normally recorded as non-vectorial graph units and space groups, which are highly dependent on domain knowledge. The feature engineering of material compounds by handmade vectorial representations that used to be the *de facto* treatment for machine learning algorithms, is not sufficient any more, because even material scientists can not easily specify key features of a material compound. When it comes to the prediction of functional properties, feature engineering in scientific discovery has always been an ad-hoc process that is dependent on domain knowledge and prone to measurement error.

In this chapter, we propose an automated approach to leverage the hidden patterns on the input end of a predictive model for materials informatics. The model is an extension of the random walk graph kernel techniques for material property prediction in which the similarity between graph objects can be automatically calculated from the non-vectorial graph representations in the material informatics databases. By constructing the graph representation efficiently from the raw geometric coordinate data using maximum spanning tree, the proposed method achieves approximately the same prediction power when compared to the conventional vectorial compound representation, and even outperforms in situations where the availability of labeled data is limited. We highlight this suitability for the current material informatics practice in which numerous potential material structures have not yet been properly annotated. Moreover, we demonstrate that the method maintains a flexible framework that enables potential inclusion of constructive domain knowledge, such as electromagnetism specified by the domain experts in materials science.

2.2 Related Work

Material informatics studies the data-driven approaches for novel material discovery. Predicting functional properties of a material compound stands at the front line of this research area. Traditional methods for predicting material properties are based on first principle computation, which is computationally inefficient and requires coarse-to-fine refinement of the model parameters based on quantum mechanics and electromagnetism [6]. With the advancements of data infrastructure and machine learning techniques, data-driven approaches for material property prediction based on machine learning techniques have achieved promising prediction accuracy with an improved computational efficiency by orders of magnitude [18][7][8]. For example, *materialsproject.org* is one of the leading databases, where 50,000+ compounds, 40,000+ band structures, 1,000+ elastic tensors, together with 2,000+ Li interclalation electrodes and 15,000+ Li conversion electrodes [18] are recorded. Compared with former databases, such as ICSD, which records data in non-exchangeable format [19], *materiasproject.org* offers rich APIs and elastic data format. This opens a new era of collective and data-driven innovations. The raw data in such databases describes the atomic species, three-dimensional polyhedra, and the *space group* related to the lattice tiling patterns for crystalline materials. However, crafting expressive features from domain-specific descriptions is non-trivial to the research community.

Several material compound representations have been proposed in literature to tackle the prediction of atomization energy, static polarizabilities, frontier orbital eigenvalues, electronic ground, and excite states of material compounds [8][9][11][28]. Coulomb matrix (CM) representation is one of this kind that has been widely used in the recent materials informatics research. The CM quantifies the electrostatic force of interaction between two point charges [10] by following one of the most essential physics laws in electromagnetism, the Coulomb's law. Many variants of the CM representation, such as eigenspectrum, sorted, and random CMs, have been proposed in order to incorporate the rotational invariance property of the atom index [11].

In principle, feature engineering requires a rational relationship between the representation and the predicted functional property of material. For example, although CM-based representations demonstrate promising results in electronic property prediction, partial radial distribution function (PRDF) representation based on the local density of different types of atoms and interactions outperforms the CM in the prediction of density of states (DOS) at the Fermi energy [9]. Although most of the former approaches focused on crafting feature vectors from the elemental unit of material compounds, an automated evaluation is demanded given the sophistication of graph-structured chemical compounds. In the current notation system, 230 space groups and a combinatorial amount of potential tiling patterns need to be considered for novel material design, while the database records only a limited number of physical properties. When the tiling patterns come into horizon, there should be more doubts than visible benefits if one use a generic feature vector to represent a material compound.

To automate the feature extraction, an alternative approach is to construct a measure of similarity or kernel between the graph objects from the raw data. A variety of graph kernels in machine learning has been proposed for different applications, ranging from computer graphics and computer vision to social networks and bioinformatics [29] [13][14][15][16][26]. Intuitively, a kernel $s : G_1 \times G_2 \to \mathbb{R}$ is a function of the similarity between two graph-structured objects. Many machine learning methods can be solved by applying the "kernel trick" in the models. For these kernel machines, a kernel implicitly control what hidden patterns count. Given a graph G defined by its vertex and edge labels, a graph kernel compares the structured objects with a hybrid of intuition and efficient computation. One example is the random walk graph kernel, which compares the matching probability of the contemporary random walk pairs in two graphs [13]. Other examples include the graphlet kernel which counts

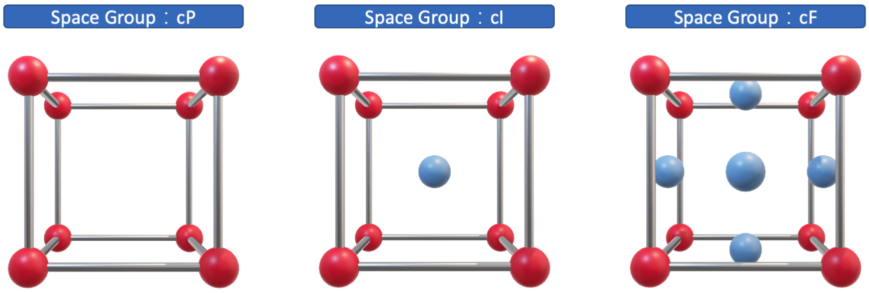


Figure 2.1: Different space group has different elemental unit and various tiling patterns of the elemental units.

the identical pairs of specific sub-graphs or sub-trees in two graphs [16]. In general, these graph kernel frameworks open a gateway for researchers to further incorporate domain knowledge into the modeling workflows and have become useful tools for domain experts [20].

2.3 Preliminaries

In material science, a crystal is structurally an infinite periodic net of *units* with an extended framework lattice using a series of basic operators such as "expansion", "decoration" and "augmentation" [30]. A *unit* can be characterized in three-dimensional geometry, such as a polyhedron. The generative process using these operators results in a large amount of potential crystals and our goal is to utilize this wide spectrum of geometric information from a unit graph as well as the lattice tiling pattern to predict functional material properties. Given a material representation $M(G,T)$, a triplet $G(V,E,L(V))$ is a vertex-labeled unit graph that shapes a polyhedron and $T \in$ *space group* is the tiling pattern of the polyhedron. The *space group* is also called the Fedorov group in crystallography. It describes the symmetry of the crystal. The vertex-labeled unit $G(V,E,L(V))$ contains the set of vertices V and the set of edges E, as well as the set of labels of the vertices $L(V)$. The vertex labels $v \in V$ are the atomic numbers of the element in the periodic table and the edges $e \in E$ are the chemical bonds interaction between the atoms. In this chapter, we consider only the isolated polyhedra, where the tiling pattern set T is empty. This is basically a simplification of the graph kernel notation for materials $M_1(G_1, T_1)$ and $M_2(G_2, T_2)$ into $s : G_1 \times G_2 \rightarrow \mathbb{R}$. Note that it is also sufficiently flexible to convert the edge-labeled graph $G_0(V_0, E_0, L(V_0))$ into an only-vertex-labeled graph $G_0(V_0^{'}, E_0^{'}, L(V_0^{'}))$ with $|V_0^{'}| = |V_0| + |E_0|$ and $|E_0^{'}| = 0$, where the number of vertices and number of edges in the graph are denoted by $|V|$ and $|E|$ respectively.

2.3.1 Random walk kernel

Given graph $G(V, E, L(V))$, a random walk kernel summarizes the similarity between joint random walks on two graphs in infinite steps. The rationality of this method lies in that, by doing joint random walks on two graphs simultaneously, two independent sequences of visited vertex labels on each graph can be generated. The similarity of two sequences generated from two random walks on respective graphs is a natural quantification of the similarity between the two graphs. In practice, by introducing a termination parameter λ, the random walk kernel quickly converges as the random walk continues; sometimes after only several iterations [27].

Assuming there are two graphs denoted by $G_1(V_1, E_1, L(V_1))$ and $G_2(V_2, E_2, L(V_2))$, a random walk on a single graph starts with an initial vertex and then transits between the vertices through the edges in the graph. Suppose that the initial distribution of the random walk position is a vector $u(0) \in \mathbb{R}^{|V|}$ and the position of the random walk at the t-th step is $u(t) \in \mathbb{R}^{|V|}$. The random walk on a single graph visits edges and vertices at intervals. At step $t = 2k-1, k = 1, 2, ..., \infty$, the random walk visits the vertices. At step $t = 2k, k = 1, 2, ..., \infty$, it visits the edges. In this way, it generates an infinite sequence of visited labels as

$$v_1, e_1, v_2, e_2, v_3, e_3...v_k, e_k...$$

The transition probability from vertices to edges $T_{v\rightarrow e}$ and the transition probability from edges to vertices $T_{e\rightarrow v} = T_{v\rightarrow e}^{'}$ are defined based on the connectivity of the vertices given in edge set

E. At each time epoch t, the random walk kernel has a termination probability $1 - \lambda$, where $\lambda \in [0, 1]$. The dynamics of the random walk are reflected by the transition matrix and is given as

$$u(t) = \lambda T u(t-1) \tag{2.1}$$

where

$$T = \begin{cases} T_{v \to e} & (t = 2k - 1) \\ T_{e \to v} & (t = 2k) \end{cases}$$

A general treatment for this transition matrix T is to assume a uniform distribution over the connected vertices and edges. For example, a fully connected three-dimensional graph with $G(V, E, L(V))$ has two transition matrices, i.e., a node-to-edge transition matrix $T_{v \to e}(t) \in \mathbb{R}^{|V| \times |E|}$ and an edge-to-node transition matrix $T_{e \to v}(t) \in \mathbb{R}^{|E| \times |V|}$, where $|E| = |V|(|V| - 1)$.

In order to compute the possibility of two labeled sequences that are generated from independently two graphs, the dynamic of the simultaneous joint random walks on G_1 and G_2 can be denoted by the Kronecker tensor product of the dynamic on both graphs as $U(t) = u(t_1) \otimes u(t_2)'$, which is identical to

$$U(t) = (\lambda_1 T_1) U(t-1) (\lambda_2 T_2') \tag{2.2}$$

By defining the label matching matrices $M_{\text{vertex}} \in \mathbb{R}^{|V_1| \times |V_2|}$, $M_{\text{edge}} \in \mathbb{R}^{|E_1| \times |E_2|}$ within elements in vertex alphabets and edge alphabets, respectively, we can derive the dynamics of a matching joint random walk,

$$U_{\text{matching}}(t) = M * (\lambda_1 T_1) U_{\text{matching}}(t-1) (\lambda_2 T_2^T) \tag{2.3}$$

where * denotes the Hadamard element-wise product of the matrices and

$$M = \begin{cases} M_{\text{vertex}} & (t = 2k - 1) \\ M_{\text{edge}} & (t = 2k) \end{cases} \tag{2.4}$$

After infinite steps of such joint random walk, the random walk graph kernel, which expresses the matching probability of two random walks, can be estimated by

$$K(G_1, G_2) = (1 - \lambda_1)(1 - \lambda_2) \sum_{t=1}^{\infty} \text{sum}(U_{\text{matching}}(t)) \tag{2.5}$$

where $\text{sum}(X)$ is the sum of all elements in matrix X.

2.3.2 Kernel machine regression

Kernel ridge regression (KRR) is a kernelized least-squares regression with L_2 regularization [12]. By applying the kernel trick, the prediction can be derived from a linear combination of the kernels between the test and training data through dual parameters $\alpha \in \mathbb{R}^n$. After an implicit construction of the graph kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ from graph-structured objects, one can substitute the defined kernel in the KRR with our pre-computed kernels. An analytical solution to this variant of KRR exists [34]; given a graph kernel $\mathbf{K} \in \mathbb{R}^{n \times n}$, the analytic solution for the dual parameters is

$$\alpha = (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{y} \tag{2.6}$$

Note that the predictions for test samples based on KRR largely rely on the computation of the kernel $\mathbf{K}(x_i, x)$ between the training set and the testing samples. Because the solution of the dual parameters is not essentially sparse, the prediction time of the testing data grows quadratically w.r.t. the size of the training set. To accelerate the prediction time of such graph

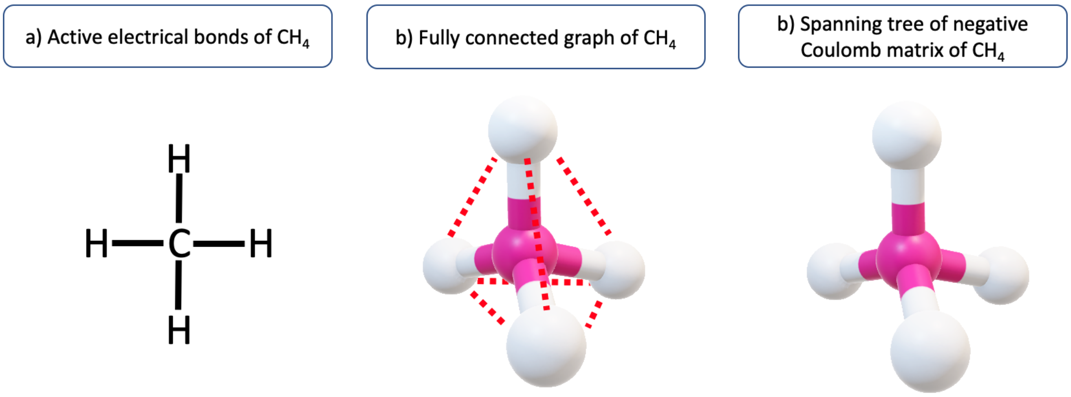

a) Active electrical bonds of CH_4
b) Fully connected graph of CH_4
b) Spanning tree of negative Coulomb matrix of CH_4
H
H
C
H
H

Figure 2.2: Graph representations of CH_4 molecule for graph kernel computation. The red atom is the carbon atom and the grey atoms are hydrogen atoms.

kernels, support vector regression can be used to solve for a sparse solution. In our experiments, the KRR is adopted, because the prediction accuracy is prioritized over the explainability.

2.4 Proposed Methods

2.4.1 Spanning tree graph representation

In material informatics databases, there is limited information describing chemical bonds, despite that all the three-dimensional coordinates of the atoms are given. To properly incorporate the chemical bonds in the predictive analysis, a naive approach is to represent the compound as a graph object G. A fully connected graph for the random walk kernel computation with a transition matrix of size $|E| \times |V|$ results in $O(|E||V|)$ computational complexity in each update of Equation (2.3). However, the fully connected graph in Figure 2(b) does not characterize the material properties well in the sense that the existence of random walk transitions between a large amount of light-weighted atoms would potentially dilute the matching signals of the sequences, especially when dominating atoms are involved.

In order to carve out the relevant active bonds from the graph-structured representation of materials compound, one may use the Cartesian coordinates of the compounds and construct a combinatorial pool of graph representations. However, only one of the candidate representations corresponds to the genuine structure that explains active bonds between atoms, as depicted in Figure 2(a). From the chemical physics perspective, a polyhedron of chemical compound should follow a heavy-atom-centered, symmetric principle for the sake of the stability of the structure. To generate an informative graph representation of the given chemical compounds systematically and automate the predictive modeling pipeline, we use the maximum spanning tree to estimate the adjacency of a valid trimmed graph from the fully connected adjacency matrix $W \in \mathbb{R}^{|V|\times|V|}$ of the chemical compound. In the adjacency matrix, $W_{i,j}$ denotes the hypothetical strength of the interaction between vertex i and vertex j. In practice, either a Coulomb matrix (CM) or a distance matrix can be used as a candidate for W in order to generate a trimmed symmetric graph of the data.

The entries in a Couloumb Matrix can be formulated by combining the Cartesian coordinates of the atoms with the corresponding nuclear charge of each atom, formulated by

$$C_{i,j} = \begin{cases} 0.5Z_i^{2.4} & (i = j) \\ \frac{Z_i Z_j}{|R_i - R_j|} & (i \neq j) \end{cases} \tag{2.7}$$

where Z_i is the nuclear charges of the i-th atom and R_i is the Cartesian coordinates of the i-th atom in the three-dimensional space. Since the CM characterizes the strength of the electronic force between atoms and is positively correlated with the chemical bonds between atoms, the adjacency matrix $A \in \mathbb{R}^{|V|\times|V|}$ generated by the minimum spanning tree on the negative CM constitutes a fine-trimmed core-rooted symmetric graph representation, as shown in Figure 2.2(c). In practice, the Kruskal's algorithm is used to ensure the heavy bonds are included in the final structure, because the Kruskal's algorithm starts from a sorting operation, which can ensure the inclusion of strong bonds and thereby the centricity of heavy atoms. To ensure that the search will start from the centralized heavy atom, one may aggregate the edge values associated with each vertex and select the highest ranked vertex as the starting point of search. This initialization works empirically in our application and its time complexity is $O(|V|\log(|V|))$.

From the overall computation complexity perspective, by taking the element-wise product of this adjacency matrix A with the original fully connected weighted matrix, which can be either a CM or a distance matrix (DM), one can obtain a graph representation G', that has a reduced number of edges at $|E'| = |V| - 1$ instead of $|E| = |V|^2$. Thereby, the computational complexity of each update in Equation (2.5) can be reduced from $O(|V|^3)$ to $O(|V|^2)$. In theory, the kernel defined in Equation (2.4) should be positive semi-definite as a valid kernel in the kernel machine. In practice, it is simply useful to have kernels defined using domain knowledge and adopt the exponential kernel to transform the similarity matrix into a positive semi-definite kernel [21].

The exact matching of labeled sequences results in a coarse evaluation of the graph kernel. However, beyond such an explicit relationship between the two graphs, domain knowledge in material science is also constructive for the prediction of material properties. Enlightened by this rich source of additional information, we propose multiple fine-grained graph kernels, i.e., vertex kernel and edge kernel in order to enrich the prediction power of the graph kernel framework. The proposed vertex kernel is a generalized evaluation of concordance of random walk sequences with a focus on the proximity of atomic elements in the infamous periodic table, as illustrated in Figure 2.3. Similarly, an edge kernel can be constructed via a similar technique that allows for incorporating the proximity of bond strength.

2.4.2 Vertex kernel with domain knowledge

In the construction of graph kernel, the primitive technique is to calculate an exact matching between the atomic numbers V_1 and V_2:

$$M^{(0)}_{\text{vertex}}(V_1, V_2) = \delta(V_1, V_2) \tag{2.8}$$

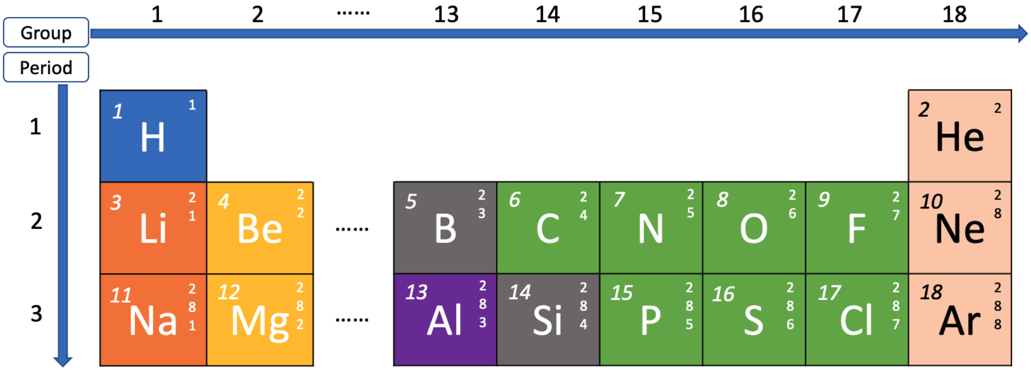


Figure 2.3: Periodic table for chemical elements, describing the atomic number (upper-left in the cubicle), energy level (the row in the periodic table), and the distribution of the electrons (upper-right in the cubicle).

where $\delta(x_1, x_2)$ is the Dirac delta function.

To incorporate a soft matching between the label sequences generated from the random walk, one can use the following Gaussian kernel to estimate the similarity between the atomic numbers in the periodic table,

$$M^{(1)}_{\text{vertex}}(V_1, V_2) = \exp\left(-\frac{(V_1 - V_2)^2}{2\sigma^2}\right) \tag{2.9}$$

where $\sigma > 0$ is the bandwidth of the kernel controlling the distance between different objects. By setting $\sigma \to 0$, Gaussian kernel is identical to the Dirac delta kernel.

Other variants of vertex kernels can be derived according to the relative position of the atoms in a periodic table. For example, the rows and columns in a periodic table are informative, describing the energy level of each atom and the free electrons in the outer orbit of the elements, respectively. These are important characteristics for determining the electromagnetism properties of the chemical compounds, as shown in Figure 2.3. Motivated by the domain knowledge in the periodic table, given the atomic number of two atoms in the compound, an energy level kernel can be defined as the following

$$M^{(2)}_{\text{vertex}}(V_1, V_2) = \exp\left(-\frac{(\text{level}(V_1) - \text{level}(V_2))^2}{2\sigma^2}\right) \tag{2.10}$$

where level(A) is the energy level of the atom in the periodic table, which corresponds to the row index of atom A in the table.

Similarly, to leverage the domain knowledge behind the number of free electrons in the outer orbit of an atom, one can define the free electron kernel as the following

$$M^{(3)}_{\text{vertex}}(V_1, V_2) = \exp\left(-\frac{(\text{free}(V_1) - \text{free}(V_2))^2}{2\sigma^2}\right) \tag{2.11}$$

where free(A) is the number of electrons in the outer orbit, corresponding to the column index of atom A in the table.

2.4.3 Edge kernel with domain knowledge

The similarity between edge labels provides another perspective to in the evaluation of kernel entries. the edge labels between the vertices can be quantified by the strength of the bond or distance between atoms. As described in Section 2.4.1, depending on whether the fully weighted matrix W is described as a Couboumb matrix or a distance matrix, the trimmed graph representation can result in different graph representations. In either case, we propose to use the Gaussian kernel to express a real-numbered strength level of the interaction between two atoms by

$$M_{\text{edge}}(E_1, E_2) = \exp\left(-\frac{(E_1 - E_2)^2}{2\sigma^2}\right) \tag{2.12}$$

where E_i is the strength of the interaction between two edges.

Table 2.1: Hyperparameters of graph kernels for kernel machine regression. ©IEEE, 2015

Name	Values
$\lambda_{\text{terminate}}$	0.1, 0.01, 0.001, 0.0001, 0.00001
σ_{edge}	0.1, 1, 5, 10, 50, 100, ∞
M_{vertex}	atomic number $M^{(1)}_{\text{vertex}}$ energy level $M^{(2)}_{\text{vertex}}$ electrons in outer orbit $M^{(3)}_{\text{vertex}}$
σ_{vertex}	0.1, 1, 5, 10, 50, 100, ∞
σ_{gaussian}	[20, 80]

2.5 Experiments

2.5.1 Dataset and experiment setting

The *qm7* dataset is used to demonstrate the effectiveness of our proposed method for the prediction of atomization energy. The dataset contains 7,165 molecules together with their compound names and Cartesian coordinates. For the *qm7* dataset, the state-of-the-art handmade feature representation is the CM representation. The range of atomization energy is from -400 to $-2{,}200$ eV/atom with its medium at $-1{,}538$ eV/atom. The alphabet of atom labels in *qm7* contains H, C, O, N, S. The largest compound in this dataset contains 23 atoms and the smallest contains 6 atoms. This information is completely correspondent with the formula of the compound without any tiling patterns inside, since it is recognized that the chemical compounds in *qm7* preferably position the heavy atoms, e.g., C, O, N, S in the core part of their polyhedron, and the light-weighted atoms, e.g., hydrogen at its periphery.

In practice, we perform coarse-to-fine parameter tuning for λ in the kernel ridge regression, starting with $10^{-10}, 10^{-9}, 10^{-8}, ..., 10^{4}, 10^{5}$, as well as the hyperparameters as shown in Table 2.1. In order to make a fair comparison of the different methods, a finer tuning of these parameters is necessary. For all the experiments, we measured the performance based on the mean absolute error (MAE) together with the variance, which implies a 70% confidence interval. We utilize the CM and the DM as two different kinds of W to generate two trimmed graph representations and compare the proposed random walk graph kernel approach with

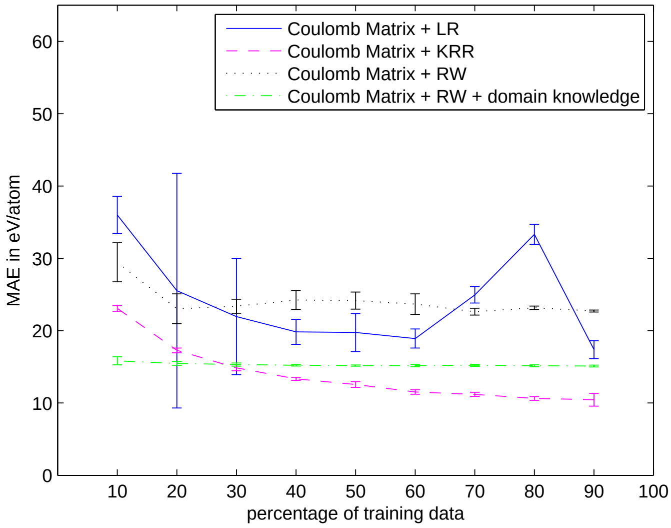


Figure 2.4: Mean absolute error when the percentage of training data increases when using Coulomb matrix representation.©IEEE, 2015

linear regression (LR), Gaussian KRR, and the random walk kernel assembled with vertex and edge kernels. We follow the 5-fold cross validation set specified in the *qm7* dataset, and determined the best hyperparameters according to the average performance of the prediction for all testing folds. When the CM is in use, we applied the sorted version of the CM as the vectorial feature of the chemical compound [11]. Similarly when the DM is in use, we applied the sorted version of the DM. The hyperparameter of the best prediction for our method is obtained at $\lambda_{\text{terminate}} = 0.00001$, $M_{\text{vertex}} \doteq M^{(1)}_{\text{vertex}}$, $\sigma_{\text{vertex}} = 1$, $\sigma_{\text{edge}} = 5$, $\lambda = 100$ and W uses the CM representation.

In Figure 2.4 and Figure 2.5, we show the test error with a 70% confidence interval by using the best model at each fraction of the training set. This provides an optimistic evaluation, because the best hyperparameter configuration may vary when the model is trained on a subset of the whole training data. Nevertheless, the performance comparison of the competing methods remains fair, even when the volume of training set is small. Further, by dividing the whole dataset into 10 equally-sized bins, we evaluate the competitiveness of the proposed method when the amount of annotation is limited. We train the competing models by gradually

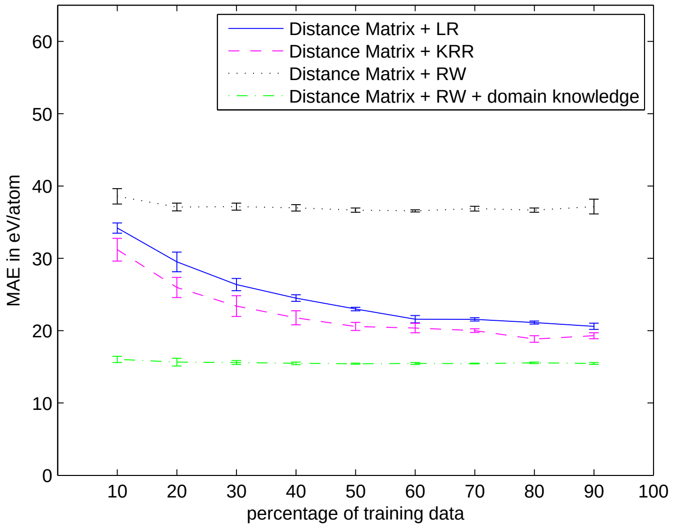


Figure 2.5: Mean absolute error when the percentage of training data increases when using distance matrix representation. ©IEEE, 2015

increasing the number of bins used for training from 10% to 90%, while using the remaining bins as test sets.

2.5.2 Results

Figure 2.4 and Figure 2.5 show the MAEs of the atomization energy prediction of the four methods that utilize the CM and the DM, respectively. In Figure 2.4, the proposed random walk graph kernel using the vertex and edge kernels, named *Coulomb matrix + RW + domain knowledge*, performs the best with a score of MAE = 15.11 ± 0.55 eV/atom. Although the *CM + KRR* also achieves a comparably good or even better score in the same 10-fold setting, the prediction performance of *CM + KRR* degenerates greatly when the training set becomes smaller. In the extreme condition, where only 10% of the whole dataset is used for training, the prediction performance of our method remains approximately the same as that of the handmade representation with 30% training data. Such an superior performance in small data setting is valid until the training set constitutes 30% of the whole dataset. In other words, we can achieve a prediction performance similar to that of the sorted CM with as little as 30% of the labeled

Table 2.2: A summary of experimental results for the competing and the proposed methods, by taking the Coulomb matrix or the distance matrix as the raw representation of material compounds. A fixed amount of 70% of dataset is used for training and validation.

Mean Absolute Error (eV/atom)		Graph representation	
Dataset	Method	Coulomb matrix	Distance matrix
qm7	mean predictor	179.02 ± 0.06	179.02 ± 0.06
	linear regression	26.18 ± 3.46	51.58 ± 4.92
	Gaussian kernel ridge regression	**10.44 ± 0.53**	33.31 ± 0.36
	RW kernel	41.24 ± 0.56	49.81 ± 1.28
	RW kernel + vertex kernel + edge kernel	**15.11 ± 0.55**	**16.35 ± 0.29**

data by the proposed method. This is suitable for the practices in novel material discovery, where a large amount of undiscovered materials needs to be accurately predicted with a limited amount of discovered (labeled) materials.

A possible reason for this is that there may exist a linear combination of the kernel matrices between the testing data and the training data through the dual parameters α. In the current dataset with 7,165 samples originally extracted from 5 specific groups [8], each 10-fold bin contains about 700+ compounds that are randomly sampled from the the 5 groups. For each bin, the calculation of a non-sparse kernelized feature vector of fixed length $\mathbf{K}(x_i, x) \in \mathbb{R}^{700+}$ is required, which is computationally heavy. This expressiveness issue greatly increases with the reduction in the training dataset for the sorted CM. In Figure 2.5, we observe a similar result for the DM. The difference between Figure 2.4 and Figure 2.5 is that the proposed graph kernel methods that take CM as an input, always outperform the sorted DM in Figure 2.5. The most competitive prediction performance is achieved by *Coulomb matrix + RW + domain knowledge*. The mean absolute errors in eV/atom for the the proposed and competing methods are summarized in Table 2.2.

2.6 Summary

In this chapter, we propose novel techniques to predict the atomization energy of molecules based on random walk graph kernel. The proposed method leverages the structured patterns and the domain knowledge in the periodic table. By trimming the graph-structured data in material informatics databases based on the maximum spanning tree algorithm, the proposed method can extract informative graph representation for graph kernel computation and reduces the computational complexity of kernel update from $O(|V|^3)$ to $O(|V|^2)$. We also demonstrate that the random walk graph kernel can be easily used to incorporate domain knowledge by using Gaussian kernel. As a promising approach to build automated predictive modeling pipeline, the proposed method demonstrates superior prediction power by achieving a near state-of-the-art prediction performance with a limited portion of labeled samples that is as little as 30% of the whole dataset. Given the fact that the availability of annotated material is limited, we argue that the proposed method is well-suited for the practices in material informatics, because it saves the simulation time by orders of magnitude, compared to the conventional quantum simulation methods in computational material science.

Chapter 3

Learning from Intransitive Relationships for Preference and Matchups

3.1 Introduction

Intransitivity is a type of data veracity that reflects the uncertainty of observations in social choice theory [81] and in preference modeling [61]. It emerges as a research topic because the data collection or generation process behind the problem setting involves uncertainty or even veracity. These issues are not properly addressed by generic models in statistics and in machine learning. For example, in ranking problems [36, 37], a top-class chess player may lose a game to a lower-ranked player, due to his under-preparation or carelessness in certain unique moves. This does not essentially mean that by observing this matchup result, one should assert that the winner of the game is surely a stronger player who will constantly win the games against the loser. Likewise, when multiple players are involved in a tournament, intransitive pairwise relationships between a group of players or objects are inevitable. In a directed graph, these intransitive pairwise relationships potentially form a cyclic preference chain, that leads us to an error-prone situation where the algorithm has to make counterfactual deterministic decisions regarding which pairwise observation to prioritize. In order to deal with these problems, the development of structured representations and learning algorithms that can handle the data veracity is demanded, despite that it is not easy to override the transitivity principle of pairwise comparison and matchup between individual objects.

Following the divide-and-concur design pattern of data structure and algorithm, escalating the dimension of representation of players raises our hope in systematically satisfying all the conflicting pairwise relationships. Given the multifaceted nature of intransitive relationship between objects, the stringency of transitivity bonded to the scalar representation for an object can be relaxed by extending the representation of the object from a scalar to 2-dimensional [52].

In this chapter, we propose a probabilistic model that joint learns the d-dimensional representation ($d > 1$) for each player and a dataset-specific metric space that systematically captures the distance metric in $\mathbb{R}^d$ in the embedding space. By imposing additional constraints in the metric space, the proposed model can degenerate to former models used in intransitive representation learning. To highlight the importance of dealing with intransitivity, an extensive quantitative investigation is provided, showing the wide existence of intransitive relationships between objects in various real-world datasets. The missing value inference experiments show that the proposed method outperforms its competing methods in terms of prediction accuracy for a variety of real-world applications, including social choice, election and online gaming.

3.2 Related Work

In literature, pairwise comparison refers to the situation where two participants are evaluated by a third-party judge or an objective rule that decides the discriminative *win/lose* result for each player. Examples of such a comparison appears in recommender systems [63], social choice systems [80, 70, 81], and so on. In pairwise matchup, two participants are each other's competitive opponents, and therefore the discriminative win/lose result is a reflection of their strength in the game. Examples of such matchup applications are sports tournaments [68] and online games [77]. In both cases, the hidden winning ability of each individual object can be quantitatively profiled by parametric probabilistic models [75].

Many existing works on parametric models for pairwise matchups data are originated from the seminal works in computational statistics and choice theory, including the Thurstone model and the Bradley-Terry-Luce (BT) model [75]. The BT model considers restrictly pairwise comparison without tie. The parameter estimation is based on maximum likelihood estimation. However, the expressiveness of the BT model is limited without further generalization in practice. To improve the expressiveness, some works have been further proposed to handle multiparty matchups and comparisons involving a tie [53]. The first generalization of BT model to multi-dimensional representation was limited to the 2-dimensional case with a non-linear logistic function, inspired by classical multidimensional scaling. In real-world matchups, the ranking of the players' ability is closely related to the parametric modeling for pairwise matchup data. For instance, in sports tournaments [49] and online games, the Elo ratings system and the TrueSkill ratings system are noteworthy. On the other hand, instead of modeling the matchups between individual players, some methods concentrate on group matchup, rating individual players from group matchup records, or alternatively model the belief of each collected record. These methods are different from ours because they all strictly assume the principle of transitivity.

In the context of modeling intransitivity, a 2-dimensional vector can be used to represent the ability of players in a matchup. The state-of-the-art model for intransitive modeling is the BC model [77], which imitates the offense and defense characteristics of a player and learns the corresponding multidimensional representations from matchup records. The BC model was then extended to context-aware settings with an improvement in the performance. Existing works in this line of research include the studies on the seminal Bradley-Terry pairwise comparison model and some extensions and applications in various real-world data science applications, e.g., matchup prediction [68], social choices [70], and so on. In the BT model, the

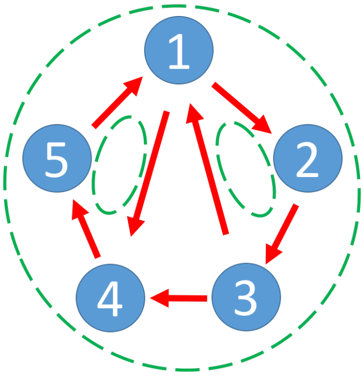


Figure 3.1: A directed asymmetric graph that illustrates the observed intrasitivity relationships in Table 3.1 ©Springer, 2017

strength of the players is parameterized as a single scalar value so that the matchups between players always remain transitive. Other attempts to meet the challenge by extending the scalar into a 2-dimensional vector representation through a non-linear logistic model. Recently, the Blade-Chest model with a multidimensional embedding scheme is proposed to imitate the offense and defense ability of a player in two independent multidimensional spaces [77]. However, the BC model, which was extended directly from the seminal BT model, is limited in its expressiveness of intransitivity due to its arbitrary separation of two representation metric spaces, which technically leads to an unfavorable numerical conjugation. In theory, a thorough theoretical justifications of these parametric probabilistic models requires certain levels of transitivity; however, the existence of *intransitivity* in the real world overrides the transitivity of preference and therefore has been widely discussed also in econometrics, behavior economics, and social choice theory for decades.

Overall, the challenge in modeling intransitivity motivates alternative approaches, such as learning intransitivity-compatible multidimensional embedding from pairwise comparisons. Without loss of generality, we denote the participants in a pairwise comparison or a matchup as *players* in our context, and discuss only the non-tie case.

3.3 Preliminaries

3.3.1 Intransitivity

Intransitivity refers to the property of binary relations (i.e., *win/loss* or *like/dislike*) that are not transitive. For instance, in a rock-paper-scissors game, the pairwise matchup result is judged by three rules: $\{o_{paper} \succ o_{rock}$, $o_{rock} \succ o_{scissors}$, and $o_{scissors} \succ o_{paper}\ \}$. A transitive model results in a transitive dominance $o_{paper} \succ o_{scissors}$, that violates the third rule $o_{scissors} \succ o_{paper}$. In other words, the binary relationships in the rock-paper-scissors game are not transitive. In the real world, such intransitivity exists in the form of cyclic dominance that implies the non-existence of a local dominant winner in the local preference loop. Moreover, the presence of a nested local intransitive preference loop results in intransitive comparison and matchup systematically; therefore predictive modeling with such data veracity is challenging. This situation occurs when objects have multiple features or views of judgment and each of these views dominates a corresponding pairwise comparison. Although an underestimation of such cyclic dominance can be subtle in the numerical testing scores in terms of prediction accuracy, it is critical in the perspective of making the learning algorithm explainable and cost-sensitive in decision making. A toy example is illustrated in Figure 3.1 and Table 3.1 to highlight the importance of learning with intransitivity.

Figure 3.1 shows a directed asymmetric graph (DAG) to illustrate the toy game records in Table 3.1; the numbered nodes represent the corresponding player, the arrows demonstrate the dominant relationship between players, and the three dotted circles demonstrate the existing cyclic intransitive dominance relationships in the observed game records. In Table 3.1, the last two columns are exemplar predictions derived from transitive and intransitive models. The prediction of a transitive model pred$_{trans}$ cannot fully capture the intrinsic intransitivity in the dataset, leading to a deterioration in terms of predictive performance, whereas the prediction made by an intransitivity-compatible model pred$_{intrans}$ is able to accurately capture all the

Table 3.1: A toy example demonstrating the subtle deterioration of test accuracy when intransitive pairwise relationships are involved in a ranking analysis. ©Springer, 2017

Winner ID	Loser ID	#wins	#loses	GT	pred$_{trans}$	pred$_{intrans}$
1	2	10	5	✓	✓	✓
1	3	1	2	✓	x	✓
1	4	10	5	✓	✓	✓
1	5	1	2	✓	x	✓
2	3	10	5	✓	✓	✓
3	4	10	5	✓	✓	✓
3	5	10	5	✓	✓	✓
4	5	10	5	✓	✓	✓
				Test Accuracy:	0.6458	0.6667

deterministic matchups. The mis-prediction of two out of the eight pairwise relationships leads to a subtle deterioration of the average test accuracy by 0.0208. However, a growth in the number of observed records may introduce a further difficulty in the evaluation of the unveiled intransitivity. In this toy example, the local intransitive sets {1,2,3} and {1,4,5} are nested in a global intransitive set {1,2,3,4,5}. Such c locally nested structures in a dataset with a large number of players n and active dominance e lead to an exponentially growing number of intransitive cycles. To enumerate the ground truth for all such cycles, the most efficient algorithm for searching all such cycles yields a time complexity bounded by $O((n+e)(c+1))$ [78], which is intractable for randomly observed dense matchups that involves a large number of participants. Therefore, brute force approaches to satisfy all the possible views are not practical. This motivates the development of novel techniques such as learning structured representation of objects from data. In the next section, we provide a quantitative exploration of cyclic intransitive relationships that are observed in a variety of real-world datasets.

3.3.2 Intransitivity in datasets

We investigate several benchmark datasets collected from diversified areas. The datasets are commonly grounded on pairwise comparisons or matchups between objects or players.

Table 3.2: Summary of intransitivity real-world datasets. ©Springer, 2017

DATASET	No. of Players	No. of Records	isIntrans	Intrans@3	No. PlayerIntrans@3
SushiA	10	100000	x	0.00%	0/10
SushiB	100	25000	✓	26.87%	92/100
Jester	100	891404	✓	1.77%	97/100
MovieLens100K	1682	139982	✓	0.19%	1130/1682
ElectionA$_5$	16	44298	✓	0.44 %	6/16
SF4$_{5000}$	35	5000	✓	23.86%	34/35
Dota	757	10442	✓	97.58%	550/757

SushiA and SushiB [70] are food preference datasets. Jester [79] and MovieLens100K [80] are collective preference datasets in an online recommender system. ElectionA$_5$ [81] is an election dataset for collective decision making. In the area of online gaming, SF4$_{5000}$ [77] is a dataset collected from professional players and is used to evaluate the strength and weakness of players. Similar to SF4$_{5000}$, Dota [77] is a dataset of game records containing a large number of players on an online RPG game platform.

A summary of the quantitative statistics of intransitive relationships in these datasets are presented in Table 3.2. *isIntrans* indicates the existence of the intransitivity relationships. *Intrans@3* indicates the percentage of intransitive loops that are analogous to the rock-paper-scissors game, where the number of involved players equals 3. In *Intrans@3*, the denominator is the total number of directed length-3 loops given by $2\binom{N}{3}$ for a fully observed pairwise dataset. *PlayerIntrans@3* is the number of players who are involved in a rock-paper-scissors-like relationship. Both *Intrans@3* and *PlayerIntrans@3* characterize the intensity of intransitivity, and a higher score indicates more intensive intransitivity in the dataset. In the majority of the seven datasets we investigated, intransitive relationship between players exists. Moreover, in five out of the seven datasets, more than half of the players are involved in local intransitive relationships. This highlights the necessity of modeling the intransitivity. The facts listed in Table 3.2 is also the first empirical exploration of the existence of intransitive relationships in the respective domain applications.

3.3.3 Bradley-Terry model

Bradley-Terry model [75, 76] is one of the most fundamental model for pairwise comparison. A critical determinant of the Bradley-Terry model is matchup matrix $\mathbf{M} \in \mathcal{R}^{N \times N}$, whose entry $M(i, j)$ indicates the comparative advantages of item i over item j. $M(i, j) > 0$ literally reads, "item i has a comparative advantage over item j." and vice versa.

In the model, each player i has a strength parameter γ_i, and the (i, j)-th element of the matchup matrix can be defined as

$$M(i, j) = \gamma_i - \gamma_j.$$

When two players i and j play a match, the winning probability $p_{ij} = \Pr(i \succ j)$ that player i wins the match is given using the matchup matrix, that is,

$$\begin{aligned} p_{ij} &= \frac{\exp(\gamma_i)}{\exp(\gamma_i) + \exp(\gamma_j)} \\ &= \frac{1}{1 + \exp\left(-(\gamma_i - \gamma_j)\right)} \\ &= \sigma(M(i, j)), \end{aligned} \tag{3.1}$$

where $\sigma(x) = 1/(1 + \exp(-x))$ is the sigmoid or logistic function.

The model has several core properties: Firstly, $M(i, j) > 0$ means i has more than 50% chance to win, and $M(i, i) = 0$ means it is an even matchup; Secondly, $M(i, j) \to +\infty$ means $\Pr(i \succ j) \to 1$; Lastly, the matchup matrix $\mathbf{M}$ satisfies the following negative symmetry condition,

$$\mathbf{M} = -\mathbf{M}^\top. \tag{3.2}$$

This negative symmetry property in Equation (3.2) is put in place to ensure $p_{ij} + p_{ji} = 1$.

As a result, each player in the Bradley-Terry model has a one-dimensional strength parameter and the probability of player i winning a match against j only depends on the relative advantage of player i over player j.

3.3.4 Blade-Chest model

To capture the multi-dimensional nature of player's strength and weakness, Blade-Chest model has been proposed to overcome the limitation of scalar γ_p that is not intransitivity-aware. The Blade-Chest model uses blade and chest vectors to imitate the offense and defense of a player. The abilities of player $p \in \mathbf{P}$ are parameterized by $\mathbf{a}_{blade}$, $\mathbf{a}_{chest} \in \mathbb{R}^d$. The corresponding matchup function is formulated by

- the Blade-Chest-Inner (BCI) embedding $M^{BCI}(a,b)$

$$M^{BCI}(a,b) = \mathbf{a}_{blade}^T \cdot \mathbf{b}_{chest} - \mathbf{b}_{blade}^T \cdot \mathbf{a}_{chest} \tag{3.3}$$

- the Blade-Chest-Distance (BCD) embedding $M^{BCD}(a,b)$

$$M^{BCD}(a,b) = \|\mathbf{b}_{blade} - \mathbf{a}_{chest}\|_2^2 - \|\mathbf{a}_{blade} - \mathbf{b}_{chest}\|_2^2$$

Overall, these formulations of the matchup function naturally ensure the symmetry property denoted in Condition (3.2) and are compatible with the scalar-valued representation of the players' strength in the BT model. The connection between these two formulations can also be evidenced under a mild condition [77]. Assembled by this multidimensional formulation, the BC model is the state-of-the-art model in both predictive modeling and representation learning for the players' intransitivity. Some general justifications of its expression power have been developed. However, there are still rooms for an improvement, because by treating the matchup matrix as an anchor in learning and inference, the algorithm may fail to explicitly explain high-order interactions between players.

3.4 Proposed Methods

We propose a novel model for learning with intransitive relationships in real-world datasets. The proposed model learns multi-dimensional intransitive representations for each object or player, e.g., items in a recommender system, tennis players in a tennis tournament, game players in online game platforms, or candidates in a political election. Different from the BC model, we focus on the joint learning of the d-dimensional representation ($d > 1$) for each player and a dataset-specific metric space that systematically captures the distance metric in $\mathbb{R}^d$ over the embedding space. The joint modeling of the multidimensional embedding representation and the metric space is achieved by involving two types of covariance matrices, one to capture the interactive battling result between two players on the metric space, and a second to capture the intrinsic strength of each player. Through an analysis of the symmetry and expressiveness of our proposed embedding formulation, we argue that the constrained optimization problem implied by the proposed multi-dimensional embedding formulation can be transformed into an unconstrained form. This reformulation results in a generic numerical solution of the proposed model by using stochastic gradient descent method [71]. Finally, we evaluate the effectiveness of the proposed method on a variety of real-world datasets, and demonstrate its superiority over other competitive methods in terms of predictive performance.

Assume that we are given a set of candidate players $\mathbf{P}$ with $|\mathbf{P}| = M$. The dataset $\mathbf{D}$ contains N pairwise matchup records $x_i(a_i, b_i) \in \{0, 1\}$, $i = [1{:}N]$, where the players a_i and b_i $\in \mathbf{P}$. An ordinal matchup record $o_a \succ o_b$ is the matchup record between player a and player b, meaning a beats b, and $o_a \prec o_b$, vice versa. The observed record $x(a, b)$ can be represented in a 4-tuple: either $x(a, b) = (a, b, 1, 0)$ meaning $o_a \succ o_b$ or $x(a, b) = (a, b, 0, 1)$ meaning $o_a \prec o_b$. The identical deterministic events can be aggregated, resulting in a collapsed dataset $\mathbf{D}^{collapse}$.

The data entry $x_{aggregate}(a, b) \in \mathbf{D}^{collapse}$ is given by 4-tuples in $x_{aggregate}(a, b) = (a, b, n_a, n_b)$, where n_a is the total count of observed event $o_a \succ o_b$, and n_b of $o_a \prec o_b$, accordingly.

The goal is to predict the result of matchups by learning multidimensional representation of the players, so that the structured representations are visible and can reflect the players' ability in multiple views better than the conventional models.

3.4.1 Generalized intransitivity model

By introducing two *transitive matrices* $\Sigma, \Gamma \in \mathbb{R}^{d \times d}$, we propose a generic formulation of the matchup function that jointly captures a d-dimensional representation ($d > 1$) for each player and a dataset-specific distance metric for the learned representation in $\mathbb{R}^d$ over the embedded dimensions. Let us assume we have a d-dimensional representation $\mathbf{a} \in \mathbb{R}^d$ for player $a \in \mathbf{P}$; then, we can formulate the generalized intransitivity embedding $M^G(a, b)$ as,

$$M^G(a, b) \quad = \quad \mathbf{a}^T \Sigma \mathbf{b} + \mathbf{a}^T \Gamma \mathbf{a} - \mathbf{b}^T \Gamma \mathbf{b} \tag{3.4}$$

where $\mathbf{a}$ and $\mathbf{b}$ are the d-dimensional representation for player a and player b, respectively, and $\Sigma, \Gamma \in \mathbb{R}^{d \times d}$ are the *transitive matrices*. The model parameters we attempt to learn are $\theta^G := \{\mathbf{a}, \mathbf{b}, \Sigma, \Gamma\}$. In the proposed formulation, the first term $\mathbf{a}^T \Sigma \mathbf{b}$ reflects the interaction between players, and the latter term $\mathbf{a}^T \Gamma \mathbf{a} - \mathbf{b}^T \Gamma \mathbf{b}$ reflects the intrinsic strength of each player. The embedding is proposed to model the pairwise preference, in which two properties should be preserved, i.e., preference symmetry and expressiveness.

We characterize the detailed properties of the proposed formulation in terms of symmetry and expressiveness in comparison with the BC model and verify that the BC model is a specialized formulation in a family of our generalized formulation.

3.4.2 Symmetry property

Different from other problems, such as link prediction in social networks, where the directed preference between items is naturally asymmetric, a matchup result between two players requires a preservation of symmetry as defined in Equation (3.2). We demonstrate that the proposed generalization of multi-dimensional intransitivity models meets this requirement.

Obviously, the two numerical computations of the first term $\mathbf{a}^T\Sigma\mathbf{b}$ and the latter term $\mathbf{a}^T\Gamma\mathbf{a} - \mathbf{b}^T\Gamma\mathbf{b}$ are independent, given randomized d-dimensional embeddings $\mathbf{a}$ and $\mathbf{b}$. Without a special design of $\mathbf{a}$ and $\mathbf{b}$, the sufficient condition to preserve the symmetry of the first term is

$$\Sigma = -\Sigma^T \tag{3.5}$$

which is difficult to regularize given the gradient $\nabla_\Sigma M^G(a, b)$:

$$\nabla_\Sigma M^G(a, b) = \mathbf{a}\mathbf{b}^T$$

However, the induced constrained optimization problem is difficult to solve. Alternatively, we devise an efficient solution which transforms the constrained optimization problem into an unconstrained optimization by reparameterizing Σ with Σ' by

$$\Sigma = \Sigma' - \Sigma'^T \tag{3.6}$$

where Σ' is a free matrix with the same shape as Σ. The symmetry of $\mathbf{a}^T\Sigma\mathbf{b}$ can be thereby satisfied via this reparameterization trivially. Together with the fact that the symmetry of the self-regulation term $\mathbf{a}^T\Gamma\mathbf{a} - \mathbf{b}^T\Gamma\mathbf{b}$ in M^G holds constantly, we conclude that the symmetry of the proposed matchup function formulation is guaranteed.

3.4.3 Expressiveness

A superior expressiveness of our proposed intransitive representation of objects can be expected by relating the proposed structured representation to the related works.

Suppose that we have blade and chest vectors for player a, $\mathbf{a}_{blade}$ and $\mathbf{a}_{chest} \in \mathbb{R}^{d'}$, where $d' = 3$; then, we integrate them into a generalized vector $\mathbf{a}_{general} \in \mathbb{R}^{2d'}$ defined by

$$\mathbf{a}_{general} = \begin{bmatrix} \mathbf{a}_{blade} \\ \mathbf{a}_{chest} \end{bmatrix} = \begin{bmatrix} blade_1 \\ blade_2 \\ blade_3 \\ chest_1 \\ chest_2 \\ chest_3 \end{bmatrix} \tag{3.7}$$

This metaphorical definition is derived from the BC model. As a result, the $2d'$-dimensional $\mathbf{a}_{general}$ has two distinct subspaces $\mathbf{a}_{blade}$ and $\mathbf{a}_{chest}$, explicitly indicating the physical strength and weakness of player a in the BC model.

Theorem 1 (Expressiveness)**.** *Given the proposed matchup formulation in $2d'$-dimensional space, the proposed model degenerates to a BCI model in d'-dimensional space, under mild condition*

$$||\mathbf{a}||_2^2 = ||\mathbf{b}||_2^2 \tag{3.8}$$

$$||\Gamma||_F \to 0$$

and,

$$\Sigma = \begin{bmatrix} 0 & I_{d' \times d'} \\ -I_{d' \times d'} & 0 \end{bmatrix}$$

Proof 1. *On the one hand, by the identified sufficient Condition (3.5) for the symmetry of* $\mathbf{a}^T\Sigma\mathbf{b}$*, given* $I_{d'\times d'}$ *as a* d'*-dimensional identity matrix, a fixed transitive matrix* Σ *with*

$$\Sigma = \begin{bmatrix} 0 & I_{d'\times d'} \\ -I_{d'\times d'} & 0 \end{bmatrix}$$

is a sufficient condition to preserve the symmetry of $\mathbf{a}^T\Sigma\mathbf{b}$*, and results in*

$$\begin{aligned} \mathbf{a}^T\Sigma\mathbf{b} &= \begin{bmatrix} \mathbf{a}_{blade} \\ \mathbf{a}_{chest} \end{bmatrix}^T \begin{bmatrix} 0 & I_{d'\times d'} \\ -I_{d'\times d'} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{b}_{blade} \\ \mathbf{b}_{chest} \end{bmatrix} && (3.9) \\ &= \mathbf{a}_{blade}^T \cdot \mathbf{b}_{chest} - \mathbf{b}_{blade}^T \cdot \mathbf{a}_{chest} && (3.10) \end{aligned}$$

On the other hand, given $\|\mathbf{a}\|_2^2 = \|\mathbf{b}\|_2^2 = c$*, the inequality* $\left\|\mathbf{a}^T\Gamma\mathbf{a} - \mathbf{b}^T\Gamma\mathbf{b}\right\| \leq 2c\,\|\Gamma\|$ *holds. Thus,* $\mathbf{a}^T\Gamma\mathbf{a} - \mathbf{b}^T\Gamma\mathbf{b} \to 0$ *holds by* $\|\mathbf{a}\|_2^2 = \|\mathbf{b}\|_2^2 = c$ *and* $\|\Gamma\|_F \to 0$.

Therefore, the BCI model can be recovered by our proposed model. □

Based on the fact that BCI formulation M^{BCI} achieves better predictive performance than its variant M^{BCD} in practice and the proposed formulation M^G is inclusive of M^{BCI}, we argue that the proposed method is superior in terms of expressiveness over the competing methods, including the BC model [75, 77].

3.5 Parameter Estimation

3.5.1 Training

Without loss of generality, given a set of players $\mathbf{P}$ and a collapsed training dataset $\mathbf{D}^{collapse}$ with pairwise matchup between players in 4-tuple (a, b, n_a, n_b), as exemplified previously, we estimate the model parameters $\theta^G := \{\mathbf{a}, \mathbf{b}, \Sigma, \Gamma\}$ so that the predictive model can better predict unseen matchups. Following Equation (3.6), we reparameterize the transitive matrix

Σ as Σ' and optimize $\theta^{G'} := \{\mathbf{a}, \mathbf{b}, \Sigma', \Gamma\}$. Similar to the BT model, we train the model by maximizing the cross-entropy objective,

$$L(D|\theta^{G'}) = \prod_{(a,b,n_a,n_b)\in\mathbf{D^{collapse}}} Pr(o_a \succ o_b)^{n_a} \cdot Pr(o_a \prec o_b)^{n_b}$$

where $Pr(o_a \succ o_b)$ is the probability of the event $o_a \succ o_b$.

In practice, we calculate the log-likelihood and optimize it with stochastic gradient descent method [71]. In each epoch of the stochastic gradient descent update, we randomly sample one 4-tuple from $\mathbf{D}^{collapse}$ and update the model parameters $\theta^{G'}$ w.r.t. the corresponding players in the tuple until convergence.

3.5.2 Regularization

The regularization terms are chosen as the following:

$$R_1(D|\theta^{G'}) = \sum_{a\in\mathbf{P}} \frac{1}{2} \|\mathbf{a}\|_2^2$$

$$R_2(D|\theta^{G'}) = \|\Sigma'\|_F$$

$$R_3(D|\theta^{G'}) = \|\Gamma\|_F$$

where $\|\cdot\|_2$ is L_2 norm and $\|\cdot\|_F$ is Frobenius norm. R_1 regularizes the scale of our embedding by intuition, as well as the scale of the blade and chest jointly, since they are integrated into our embedding. R_2 regularizes the scale of the free matrix Σ' as well as the scale of the symplectic matrix Σ, because $\|\Sigma\|_F = \left\|\Sigma' - \Sigma'^T\right\|_F$ is upper bounded by $2\|\Sigma\|_F$. R_3 regularizes the scale of the free matrix Γ, in line with Condition (3.8) given in Theorem 1.

Overall, the regularized training objective for a given training dataset is

$$Q(D,\theta^{G'}) = L(D|\theta^{G'}) - \sum_i \lambda_i R_i(\theta^{G'}) \tag{3.11}$$

where $\theta^{G'} := \{\mathbf{a}, \mathbf{b}, \Sigma', \Gamma\}$ denotes the model parameters and λ controls the regularization.

3.6 Experiments

3.6.1 Evaluation metric

Cross validation for parameter tuning is used in the missing value inference experiments. Given the dataset in 4-tuple format, we first split the dataset randomly into three folds for cross validation and then identify the unique pairwise interactions and aggregated the interactions before initializing the model parameters. The hyperparameters are the dimensionality of the embedding d and the regularization coefficient λ. The performance is measured by the average test accuracy $A(\mathbf{D}_{test}|\theta)$, defined by

$$A(\mathbf{D}_{test}|\theta) = \frac{1}{|\mathbf{D}_{test}|} \sum_{(a,b,n_a,n_b)\in\mathbf{D}_{test}} n_a \cdot \mathbb{1}(\hat{o}_a \succ \hat{o}_b) + n_b \cdot \mathbb{1}(\hat{o}_a \prec \hat{o}_b)$$

where $\mathbb{1}(\cdot)$ is the indicator function of an event.

The proposed method is compared with three competitive methods, namely the naïve method, BT model, and BC model. The **naïve method** estimates the winning probability of each player based on the empirical observations, with $Pr(o_a \succ o_b) = \frac{n_a+1}{(n_a+1)+(n_b+1)}$. If $n_a = n_b$, one player is randomly assigned as the winner. The **BT model** estimates player ability with a scalar representation. The **BC model** estimates player ability with two multidimensional vectors that are independent of each other.

Table 3.3: A summary of test accuracy on datasets in food preference, recommender system and online gaming. ©Springer, 2017

DATASET	Naïve	Bradley-Terry	Blade-Chest	Proposed Model
SushiA	0.6549 ± 0.0044	0.6549 ± 0.0021	0.6551 ± 0.0038	$\mathbf{0.6551 \pm 0.0027}$
SushiB	0.6466 ± 0.0042	0.6582 ± 0.0077	0.6591 ± 0.0051	$\mathbf{0.6593 \pm 0.0058}$
Jester	0.6216 ± 0.0006	0.6236 ± 0.0028	0.6242 ± 0.0035	$\mathbf{0.6243 \pm 0.0019}$
ElectionA$_5$	0.6507 ± 0.0031	0.6531 ± 0.0038	0.6533 ± 0.0043	$\mathbf{0.6535 \pm 0.0055}$
SF4$_{5000}$	0.5297 ± 0.0102	0.5329 ± 0.0044	0.5329 ± 0.0062	$\mathbf{0.5355 \pm 0.0080}$

3.6.2 Results

Table 3.3 shows the experimental results of our proposed method. For all of the four transitivity-rich datasets, SushiB, Jester, ElectionA$_5$, and SF4$_{5000}$, we observe improvement in terms of the average test accuracy. In addition to the predictive performance, two practical facts are noteworthy. (a) The observed pairwise interactions in all these datasets are rich, and a K-fold cross validation procedure with no data augmentation results in a set of data bins, each of which contains identical players. As a result, it is guaranteed that the representation of each player in the validation and test bin is learned in the training phase that utilizes $K-2$ bins. However, as the number of players grows, the number of records required to accommodate such a full-evidenced cross validation procedure grows quickly. For instance, in the case of the SushiB dataset with 100 players, 25000 pairwise records, Intrans@3 $= 26.87\%$, and PlayerIntrans@3 $= 92/100$, the empirical down sampling for 3-fold cross validation is sufficient to perform a sufficiently evidenced prediction of the dominance for all possible player pairs, instead of a random guess caused by the existence of non-observed players in the validation and test bins. (b) Given a sampling scheme that is sufficiently stable to allow the model to give a sufficiently evidenced prediction, a K-fold cross validation results in sparser interactions in the bins, which can be indicated by the connectivity of the matchup network, i.e., Borda count or Copeland count for directed graphs. However, in the challenging MovieLens100K and Dota datasets, the resultant heterogeneous interactions between players prevent us from providing evidenced dominance prediction from the observed sparse networks.

A trivial solution for such a case is a random guess, which is meaningless for intransitivity recovery. The above two facts hold for all the competitive methods.

3.7 Summary

In this chapter, we study the issue of modeling intransitivity and representation learning for players involved in pairwise interactions. We proposed a generalized embedding formulation for learning structured representation that is compatible with intransitivity from pairwise comparison and matchup data. We also discussed the key properties of the proposed structured representation in terms of symmetry and expressiveness constructively. For the parameter estimation, an efficient solution to the constraint optimization problem involving multiple model parameters was developed based on alternating direction methods of multipliers. Through missing value inference experiments on datasets collected from recommender system, food preference and online gaming platform, we demonstrated that the proposed method improves the prediction accuracy of existing models. Moreover, we provided an extensive investigation of the wide existence of intransitive relationships between objects and players in real-world datasets. This highlights the importance of dealing with intransitivity in predicting with structured data, especially when ranking perspective is involved.

Chapter 4

End-to-end Learning from Intransitive Relationships with Structured Neural Networks

4.1 Introduction

In the previous chapter, we showed how multi-dimensional representation of objects for pairwise comparison and matchup can help in dealing with intransitivity. The simplest case is that when the intrinsic dimension of the problem is one, the problem degenerates into the Bradley-Terry model, in which no intransitivity of pairwise relationship is allowed. In order to motivate a structured discovery of hidden patterns behind the conflicting observations, two aspects of a predictive model demand further deliberation. Firstly, to preserve the generalization power of a predictive model, one may specify an intrinsic dimension d beforehand, despite that identifying the intrinsic dimension of a problem is difficult. Secondly, in order to build models with richer expressiveness to handle the conflicting observations, one can utilize deep neural networks and tailor the network architecture so that it is compatible with intransitive relationships. In this chapter, we discuss the construction of deep neural networks for the problem of learning from intransitive relationships.

The modeling of pairwise comparisons and the modeling of ranking lists share common elements, such as the interactions between players and objects. Representative pairwise comparisons are matchups, including online video games [67, 66] and sports tournaments [68], where the model is used to either pair an equal and fair game or to predict winner by systematically taking into account of overall track records. In ranking problems where rankings can be aggregated from partially observed results, pairwise comparisons and preferences are easily found as essential elements [61, 64].

The advantage of highlighting the pairwise comparisons for the ranking problem comes in two directions. Firstly, the pairwise comparison is the basic element for the data-driven model and the learning system to read and to learn from, in order to predict a *win/lose* result of future competitions. Secondly, the interpretation of the model can be streamlined in the lens of a simple matchup matrix because the matchup matrix is an abstraction that carves out the predictive ranking list.

Two of the most widely used models of ranking from pairwise comparisons are the Thurstone model [65] and the Bradley-Terry (BT) model [75, 76], both of which belong to an extended class of Random Utility Models and share a common characteristic that a scalar score is used to represent a player's overall capability. In more recent works [62], the BT model becomes the basis of all probabilistic approaches and shows a well-balanced status between a model's interpretability and its predictability.

A critical limitation of the BT pairwise comparison model is that the strength or the competitiveness of a player is modeled by using only a single scalar. In fact, it is not sufficient to represent one player with only a single scalar. It is straightforward to show that if player A beats B and B beats C more often than not, then the model will predict that A always beats C. However, intransitive relationship naturally exists in many real world applications such as economics, sports games and social choice theory. An exemplar case is a basketball tournament with three teams, i.e. team A, team B and team C. The transitive model can not satisfy the following potential outcomes at the same time: Pr(A beats B) > 0.5, Pr(B beats C) > 0.5 and Pr(C beats A) > 0.5, where Pr(A beats B) > 0.5 indicates that the probability of team A beats team B is larger than 0.5. Besides, some decisive factors, like injuries, teamwork and psychological factors, are neither explicitly included. Therefore, to improve the prediction accuracy in the inference of missing observations as well as the fairness in player pairing, innovative solutions are needed.

To deal with these problems, several models of intransitive comparison have been proposed, such as the 2-dimensional vector representation BT model [59], the Blade-Chest (BC) model [73, 77]. The BC model utilizes two multi-dimensional vectors, i.e. 'blade' and 'chest', to reflect different aspects of each player's strength. The blade and chest vectors can be then integrated into a generalized intransitive model [72], which enhances the numerical interactions between the vectors. Most of these works considered a multi-dimensional representation of each player and adopted stochastic gradient method to optimize a regularized cross-entropy objective.

In this chapter, we propose an end-to-end approach to develop an intransitive model. The model is a generalized from the Blade-Chest model, but avoids the estimation of the 'blade' and 'chest' vectors of each player by using structured deep neural networks. We show constructively how the proposed framework unifies a series of the existing works. The framework opens a gateway to future development of more expressive network structures, thanks to the automatic differentiation of deep neural network modules. Through similar experimental setting in the previous chapter, we demonstrate that our proposed method achieves a higher prediction accuracy in missing value inference experiments.

4.2 Related Work

The Thurstone [65] and the Bradley-Terry-Luce [75, 76] form the fundamentals of matchup and pairwise comparison. They use one-dimensional embedding which assumes that each item can be completely represented by an inherent strength. These work were surveyed extensively [53, 62] and the applications of utilizing such single scalar to measure the strength of the players and to estimate the ranking have expanded to wider domains, such as the matchmaking for online games [58, 54], sports [49, 55, 50] and rating system [56, 57, 45].

Multi-dimensional representation emerged in [51, 62] to advance the oversimplified probabilistic model without explicitly taking the intransitivity issue into consideration. [59] uses a 2-dimensional vector to model the property of each player, but only with verification on small datasets. [46] adopts matrix factorization to predict the scores of professional basketball games. These works do not explicitly address the intransitive property. Nevertheless, they have a common mindset of attempting to reparameterize the probabilistic model. Since then, the idea of using multi-dimensional representation for pairwise comparison has been adopted in applications such as the recommendation system [46, 47], language modeling [44].

Learning from intransitive relationships has been addressed in the data mining research community [77, 72, 83]. An example is the rock-paper-scissor in which the existence of intransitivity is authorized by the game rules. [73, 77, 72, 83] propose state-of-the-art intransitive models assembled with multi-dimensional features to differentiate the analogous plays. Context-aware models have also been developed with special concerns on feature engineering [73] and domain adaptation. [84] addresses the time-varying property of pairwise comparison by leveraging the covariance function of continuous-time Gaussian processes. [85] proposes a context dependent random utility model, which allows for a particular class of choice set effects that can oversee the prediction in the panorama. In general, incorporating context information helps improve predictive performance [39–41] and an awareness of intransitivity in decision making helps improve overall utility [42, 43].

4.3 Preliminaries

The notations in this chapter generally follow the notations defined in Chapter 3, we focus on modeling comparisons between two players and we assume the result of each matchup cannot be a draw. Given a set of players P with $|P| = N$ and a dataset $\mathcal{D}$ which contains n pairwise matchup records (i, j), where $i, j \in P$. For any players $i, j \in P$, we denote $i \succ j$ when player i wins a match against player j. For an observed matchup between each pair of players can be described in 4-tuple as $(i, j, 1, 0)$ that means $i \succ j$, or $(i, j, 0, 1)$ which means $j \succ i$. In any subset of $\mathcal{D}$, the game between the same players can be aggregated, and also resulting in a 4-tuple as (i, j, n_i, n_j). In such a tuple, n_i indicates the number of times i wins j and n_j means the opposition. In this section, we briefly review the preliminary works, including the Blade-Chest (BC) model, which allows intransitive ordering by generalizing the BT model and the Blade-Chest-Sigma model, which is a generalized from the BC model.

4.3.1 Generalized Blade-Chest model

A critical limitation of the Bradley-Terry model is that it assumes transitive relations among players; that is, if player $\gamma_i > \gamma_j$ (i.e., i has an advantage over player) and $\gamma_j > \gamma_k$, then $\gamma_i > \gamma_k$ holds. In other words, all players are transitively ordered. Such assumption may not hold in practice and a simplest counter example is Rock-Paper-Scissors game where the rule is Paper $\succ$ Rock, Rock $\succ$ Scissor and Scissor $\succ$ Paper. By sampling pairwise observation from the predefined rules, the intransitivity issue arises. It can defined as the following:

Definition 4.3.1. *Matchup relations of n players contain (stochastic) intransitivity if there exist three players i, j and k such that*

- $\Pr(i \succ j) > 0.5$*;*
- $\Pr(j \succ k) > 0.5$*;*

- $\Pr(k \succ i) > 0.5$.

Chen and Joachims [77] proposed the Blade-Chest model that allows intransitive relations among players by introducing two extra d-dimensional vectors for each player i: $\mathbf{x}_i^{\text{blade}}$ and $\mathbf{x}_i^{\text{chest}}$. The matchup matrix of the Blade-Chest model[1] is given as

$$M_{ij}^{(Blade-Chest-Inner)} = {\mathbf{x}_i^{\text{blade}}}^\top \mathbf{x}_j^{\text{chest}} - {\mathbf{x}_j^{\text{blade}}}^\top \mathbf{x}_i^{\text{chest}} + \gamma_i - \gamma_j. \tag{4.1}$$

The Blade-Chest model is a multi-dimensional extension of the Bradley-Terry model. The multi-dimensional representation in the BC model allows intransitive relations among players. Generally, the more dimensions the representation has, the more intransitivity the model allows.

A neural network framework of the Blade-Chest model has been proposed by Chen and Joachims [73]. The top layer of the neural network expresses the blade-chest-inner model in Equation (4.1). The bottom layer is a fully-connected feed-forward mapping that entangles the blade/chest vectors and feature vectors. There are two variants, namely CONCAT and SPLIT. Both variants require feature vectors that describe the environment and the player as input information.

4.3.2 Blade-Chest-Sigma model

In the previous chapter, we proposed a generic formulation of the BC model. By assuming a $2d$-dimensional representation $\mathbf{x}_i \in \mathcal{R}^2d$ for player i, the matchup matrix is given by

$$M_{ij}^{(Blade-Chest-Sigma)} = \mathbf{x}_i^\top \Sigma \mathbf{x}_j + \mathbf{x}_i^\top \Gamma \mathbf{x}_i - \mathbf{x}_j^\top \Gamma \mathbf{x}_j, \tag{4.2}$$

[1] More precisely, this model is called the Blade-Chest-Inner model, and they also propose another variant called the Blade-Chest-Dist model; however, there is no significant difference between them. And in practice, the Blade-Chest-Inner always achieves better prediction than Blade-Chest-Dist. Then we focus on the former in this paper.

where $\Sigma, \Gamma \in \mathcal{R}^{2d \times 2d}$ are the transitive matrices, $\mathbf{x}_i^\top \Sigma \mathbf{x}_j$ reflects the interaction between players and $\mathbf{x}_i^\top \Gamma \mathbf{x}_i - \mathbf{x}_j^\top \Gamma \mathbf{x}_j$ reflects the intrinsic strength of each player. We denote this model as "Blade-Chest-Sigma". Following the Theorem 1 in Chapter 3, if we initialize the matrix Σ properly as

$$\Sigma = \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{pmatrix},$$

the Blade-Chest-Sigma model in Equation (4.2) can be reduced to the Blade-Chest-Inner.

4.4 Proposed Method

4.4.1 Generalized intransitivity model

We first describe a generalized representation for the matchup matrix in the Blade-Chest model. Let us denote the representations of player i by

$$\mathbf{x}_i = \begin{pmatrix} \mathbf{x}_i^{\text{blade}} \\ \mathbf{x}_i^{\text{chest}} \end{pmatrix},$$

and define the representation matrix as

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}^{\text{blade}} \\ \mathbf{X}^{\text{chest}} \end{pmatrix},$$

where

$$\mathbf{X}^{\text{blade}} = (\mathbf{x}_1^{\text{blade}}, \ldots, \mathbf{x}_N^{\text{blade}}),$$
$$\mathbf{X}^{\text{chest}} = (\mathbf{x}_1^{\text{chest}}, \ldots, \mathbf{x}_N^{\text{chest}}).$$

We rewrite the strength parameters in the original Bradley-Terry model as

$$\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_N).$$

By using the above notations, we can see the Blade-Chest-Inner model in Equation (4.1) can be expressed as

$$\begin{aligned} \mathbf{M} &= \mathbf{X}^\top \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{pmatrix} \mathbf{X} + \boldsymbol{\gamma}^\top \mathbf{1} - \mathbf{1}^\top \boldsymbol{\gamma} && (4.3) \\ &= \mathbf{X}^{\text{blade}^\top} \mathbf{X}^{\text{chest}} - \mathbf{X}^{\text{chest}^\top} \mathbf{X}^{\text{blade}} + \boldsymbol{\gamma}^\top \mathbf{1} - \mathbf{1}^\top \boldsymbol{\gamma}. && (4.4) \end{aligned}$$

By replacing the matrix product $\mathbf{X}^{\text{blade}^\top} \mathbf{X}^{\text{chest}}$ by a new matrix $\mathbf{Y}$ defined as

$$\mathbf{Y} = \mathbf{X}^{\text{blade}^\top} \mathbf{X}^{\text{chest}}$$

we obtain a generalized representation of the matchup matrix with a rank constraint

$$\mathbf{M} = \left(\boldsymbol{\gamma}^\top \mathbf{1} - \mathbf{1}^\top \boldsymbol{\gamma}\right) + \left(\mathbf{Y} - \mathbf{Y}^\top\right) \quad \text{s.t.} \;\; \text{rank}(\mathbf{Y}) \leq D. \tag{4.5}$$

Depending on the intransitivity level requires, the structure of model can be change to deal with the increasing rank of $\mathbf{Y}$. In practice, an arbitrarily matchup matrix by $\mathbf{Y} - \mathbf{Y}^\top$ with rank D can be sufficient in terms of expressiveness, implied by the constructive example [77, Theorem 1].

4.4.2 Properties

Similar to the discussion in Chapter 3, the matchup matrix should satisfy several properties.

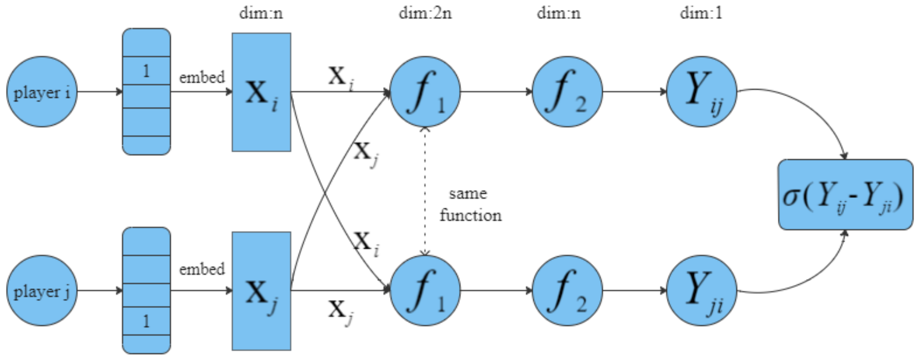


Figure 4.1: An illustration of the proposed generalized intransitivity framework. ©IEEE, 2020

Firstly, it has been mentioned above that $\mathbf{Y} - \mathbf{Y}^\top$ can represent an arbitrarily complex matchup matrix by removing the rank constraint.

Secondly, it is important to notice that the model can still represent the intransitivity even when we reduce the rank of $\mathbf{Y}$ to 1. If we take $\mathbf{Y}$ as a rank-1 matrix by

$$\mathbf{Y} = (x_1^{\text{blade}}, x_2^{\text{blade}}, \ldots, x_N^{\text{blade}})^\top (x_1^{\text{chest}}, x_2^{\text{chest}}, \ldots x_N^{\text{chest}}),$$

Thereby, the matchup matrix without the strength terms $\boldsymbol{\gamma}^\top \mathbf{1} - \mathbf{1}^\top \boldsymbol{\gamma}$ becomes

$$M_{ij} = x_i^{\text{blade}} x_j^{\text{chest}} - x_i^{\text{chest}} x_j^{\text{blade}}.$$

Assuming that $i \succ j$ and $j \succ k$ (i.e., $M_{ij} > 0$ and $M_{jk} > 0$), one can verify that the intransitive relations among three players in the rock-paper-scissors game can be expressed.

Lastly, if $\mathbf{Y} = \mathbf{0}$ and $\text{rank}(\mathbf{Y}) = 0$, the model with a separated strength terms in Equation (4.5) is equivalent to the primitive Bradley-Terry model.

4.4.3 Parameter estimation with structured neural network

Figure 4.1 illustrates the entire parameter setup and the structure of neural network that expresses the Equation (4.5). The proposed framework is essentially simpler than the framework of the Blade-Chest model in [73], because the “blade” and “chest” vectors are implicitly embedded in the model via the embedding modules.

Similar techniques that attempt to structure neural networks to solve domain specific optimization problems can be also found in the development of neural networks for model predictive control [38]. For a pairwise (i, j), we have two 0-1 vectors that encode the players’ identities. The bottom is an embedding that link the player vector into an embedded vector. We assume that there are links between the embedding vectors and the entries in the matchup matrix Y is calculated by a linear transformation

$$\begin{aligned} Y_{ij} &= A \begin{pmatrix} \mathbf{x}_i \\ \mathbf{x}_j \end{pmatrix} + b, \\ Y_{ji} &= A \begin{pmatrix} \mathbf{x}_j \\ \mathbf{x}_i \end{pmatrix} + b. \end{aligned} \tag{4.6}$$

In this linear transformation, the rank of matrix $\mathbf{Y}$ is identical to the dimension of the one-hot representation of players. Besides, the negative symmetry of the matchup matrix can be easily verified by

$$\begin{aligned} M_{ij} = Y_{ij} - Y_{ji} &= A \begin{pmatrix} \mathbf{x}_i \\ \mathbf{x}_j \end{pmatrix} - A \begin{pmatrix} \mathbf{x}_j \\ \mathbf{x}_i \end{pmatrix}, \\ M_{ji} = Y_{ji} - Y_{ij} &= A \begin{pmatrix} \mathbf{x}_j \\ \mathbf{x}_i \end{pmatrix} - A \begin{pmatrix} \mathbf{x}_i \\ \mathbf{x}_j \end{pmatrix}, \end{aligned}$$

By letting $A = \begin{pmatrix} B \\ C \end{pmatrix}$ and $b = 0$. The model can be viewed as a linear transformation of blade and chest vectors as discussed in the previous chapter.

Instead of using a logistic function to construct the cross-entropy objective, we use a fully-connected nonlinear neural network layer with an activation function f. The first hidden layer is given as

$$\begin{aligned} z_{11} &= f_1(\mathbf{x}_i, \mathbf{x}_j), \\ z_{12} &= f_1(\mathbf{x}_j, \mathbf{x}_i). \end{aligned}$$

The winning probability of the pairwise relationship between player i and player j can be estimated by

$$p_{ij} = \sigma(M(i,j)) = \sigma(Y_{ij} - Y_{ji}). \tag{4.7}$$

4.5 Experiments

The proposed model is evaluated on a synthetic dataset and several real-world datasets, covering domains such as food preference, recommender system and online gaming. The evaluation is focused on the accuracy of the predictive model in missing value inference setting.

4.5.1 Experiment setting

The input of all experiments includes: a set of players $P = \{1, 2, ..., N\}$; a dataset $\mathcal{D}$ of all the match results between player i and j in P, for any $i, j \in P$; n_i being the number of times player i win j; n_j being the opposition. The output of the experiments is a estimated strength matrix $\mathbf{Y}$.

By assuming player i is the winner, the objective function is

$$\text{argmax}_{\Theta}\ \Sigma_{(i,j)\in\mathcal{D}} \log \Pr(i \succ j|\Theta).$$

For a test partition $\mathcal{D}'$, the test accuracy is defined as

$$A(\mathcal{D}'|\mathbf{Y}) = \frac{1}{N'} \sum_{(i,j)\in\mathcal{D}} \mathbf{1}(i \succ j),$$

where N' is the total number of games in the testing set, and $\mathbf{1}(\cdot)$ is the indicator function.

We compared our proposed method with four competitive methods as following

1. Bradley-Terry model with the stochastic gradient method (BT model);

2. Blade-Chest-Inner with the stochastic gradient method (Blade-Chest-Inner);

3. Blade-Chest-Sigma with the stochastic gradient method (Blade-Chest-Sigma);

4. Blade-Chest-Inner with neural network framework (Neural BC).

We follow the same setting for BT model, Blade-Chest-Inner and Blade-Chest-Sigma with the stochastic gradient method in [77, 72]. The optimization objective is cross-entropy combined with corresponding regularization. The model includes three parameters: the parameter λ of the regularization term, the learning rate r and the embedding dimension d. We do grid search over powers of 10 from 10^{-5} to 10^{2} for λ and r, and also take $d = \{2, 5, 10, 50, 100\}$. For the Neural-BC and the proposed method, we take the batch size from 16 to 1,024 and the middle dimension from 8 to 512.

4.5.2 Results on synthetic dataset

We randomly generate the datasets which have the full intransitivity with different ranks and sizes. Let $R = \{1, 3, 5, 7\}$ be a set of ranks assumed to be the genuine dimension of the blade and chest vectors, and $S = \{500, 1000, 2000\}$ be a set of training data size. Given $N = 100$ players, for every rank element $r \in R$, we randomly generate blade and chest parameters

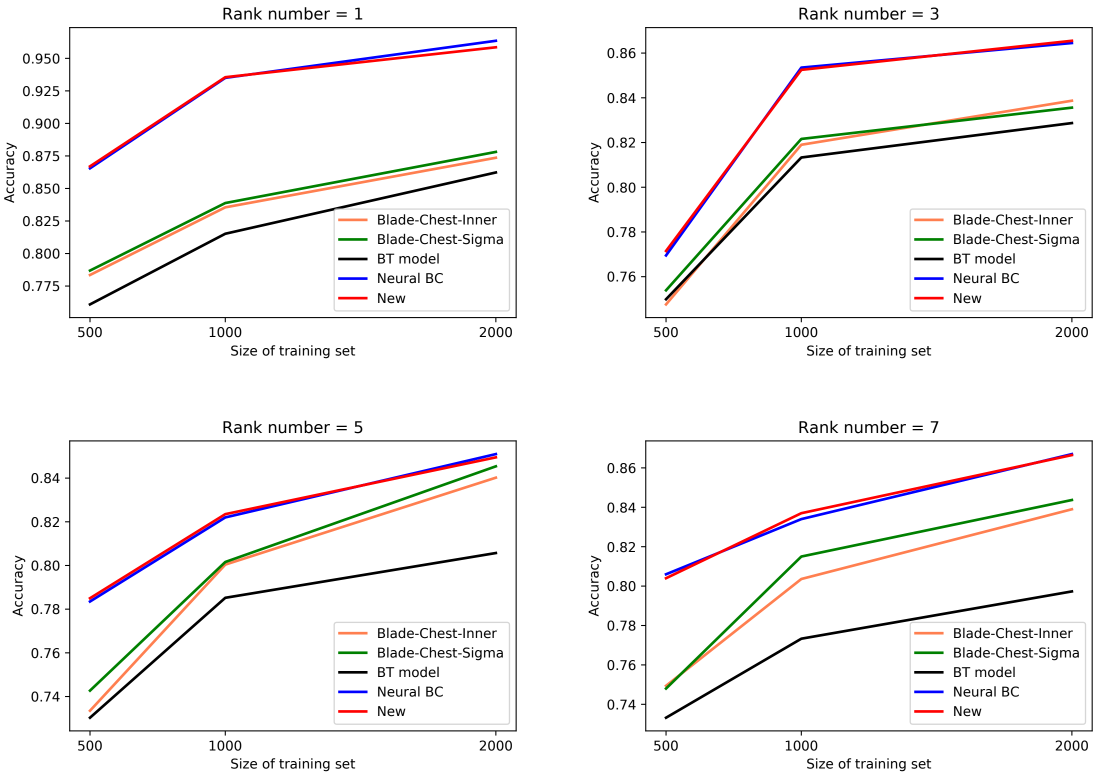


Figure 4.2: Test accuracy on the synthetic datasets with controlled rank. ©IEEE, 2020

$\mathbf{X}^*_{\text{blade}} \in \mathcal{R}^{r\times 100}$, $\mathbf{X}^*_{\text{chest}} \in \mathcal{R}^{r\times 100}$ and calculate the relation matrices $\mathbf{Y}^*$, $\mathbf{M}^*$ by

$$\mathbf{Y}^* = {\mathbf{X}^*_{\text{blade}}}^\top \mathbf{X}^*_{\text{chest}}, \quad \mathbf{M}^* = \mathbf{Y}^* - {\mathbf{Y}^*}^\top.$$

For all sizes $s \in S$, we randomly generate a training data T with size s, a validation data V with size $2,000$ and a evaluation data E with size $2,000$ using the true relation matrix $\mathbf{M}$.

The models are trained for all four rank groups and the accuracy results are averaged over 10 trials, as shown in Figure 4.2. For each rank, there are three different training sets, and we generate 10 trials for each of them.

Figure 4.2 shows the curves of the accuracy values with respect to the size of training set. For all the competitive methods, the predictive accuracy is high around 0.9 when the amount of observations becomes larger. The Blade-Chest-Inner and the Blade-Chest-Sigma perform

Table 4.1: Summary of the datasets. ©IEEE, 2020

Dataset	Players	Records	Intrans.	No.IntPlayer	Int.Ratio
SushiA	10	100000	no	0	0
SushiB	100	25000	yes	92	26.87%
MovieLens100K	1682	139982	yes	1130	0.19%
Election A5	16	44298	yes	6	0.44%
Election A9	12	95888	yes	5	1.82%
Election A17	13	21037	yes	8	8.18%
Election A48	10	25848	no	0	0
Election A81	11	44298	yes	5	2.50%
SF4-5000	35	5000	yes	34	23.86%
Dota	757	10442	yes	550	97.58%
Pokemon	800	50000	yes	784	78.58%

consistently better than the primitive BT model, while the proposed model outperforms all of them. In comparison with the Neural BC model, the proposed method performs on par in terms of predictive accuracy but with a simpler construction that drops the requirement of solving for ”blade” and ”chest” vectors.

4.5.3 Results on real-world datasets

Similar to the previous chapter, we conduct missing value experiment on datasets that include food preference datasets are SushiA and SushiB [70], online recommender system for pairwise preference datasets: MovieLens100K [60], decision making datasets: Elections [69], and online gaming: SF4, Dota [77] and Pokemon.

As illustrated in Table 4.1, the SushiA and Election A48 datasets have no intransitivity; MovieLens100K and other Elections have lower ratios of intransitivity, while SushiB and online games have a higher intransitivity ratios. Intrans. means the existence of the intransitivity, No.IntPlayer indicates the number of players those involved in rock-paper-scissors relationship, and the Int.Ratio is the percentage of intransitive loops in the whole games. We use 50% observed records for training, 20% records for validation and the remaining 30% records for

Table 4.2: Summary of test accuracy. ©IEEE, 2020

Dataset	Bradley-Terry	Blade-Chest-Inner	Blade-Chest-Sigma	Neural BC	Proposed model
SushiA	0.6525 ± 0.0011	0.6546 ± 0.0006	0.6560 ± 0.0004	0.6630 ± 0.0004	**0.6632 ± 0.0003**
SushiB	0.6257 ± 0.0025	0.6235 ± 0.0150	0.6414 ± 0.0019	0.6561 ± 0.0017	**0.6563 ± 0.0011**
MovieLens100K	0.6785 ± 0.0005	0.6792 ± 0.0004	0.6789 ± 0.0003	0.6950 ± 0.0019	**0.6973 ± 0.0002**
Election A5	0.6478 ± 0.0017	0.6489 ± 0.0011	0.6494 ± 0.0018	0.6550 ± 0.0030	**0.6560 ± 0.0018**
Election A9	0.6028 ± 0.0003	0.6096 ± 0.0007	0.6047 ± 0.0008	0.6174 ± 0.0003	**0.6175 ± 0.0003**
Election A17	0.5189 ± 0.0001	0.5305 ± 0.0010	0.5296 ± 0.0013	0.5582 ± 0.0003	**0.5598 ± 0.0002**
Election A48	0.5993 ± 0.0001	0.6001 ± 0.0001	0.5996 ± 0.0001	**0.6060 ± 0.0001**	0.6056 ± 0.0001
Election A81	0.6013 ± 0.0001	0.6018 ± 0.0001	0.6011 ± 0.0002	0.6194 ± 0.0001	**0.6194 ± 0.0001**
SF4-5000	0.5079 ± 0.0078	0.5181 ± 0.0171	0.5358 ± 0.0049	**0.5514 ± 0.0008**	0.5496 ± 0.0021
DotA	0.6334 ± 0.0077	0.6432 ± 0.0034	0.6420 ± 0.0051	0.6468 ± 0.0031	**0.6485 ± 0.0025**
Pokemon	0.8157 ± 0.0094	0.8495 ± 0.0016	0.8187 ± 0.0168	0.8943 ± 0.0040	**0.8949 ± 0.0021**

testing in each dataset by randomly separating. We do this random sampling for 3 times and report the mean and standard deviation of prediction accuracy on the test set. A summary of the averaged test accuracy of all the comparison methods is reported in Table 4.2.

Compared with the baseline methods such as the BT model, Blade-Chest-Inner, Blade-Chest-Sigma, and Neural BC, the proposed model achieved sound improvement on the test accuracy on the majority of the datasets. We attribute this advancement to the deep numerical conjugation of networked information in the neural network.

Bringing together Table 4.1 and Table 4.2, we observe that there is a high correlation between the Intransitivity Ratio in 4.1 and the variance of test accuracy in all corresponding columns in 4.2. For datasets whose Intransitivity Ratio is below 10%, e.g., SushiA, MovieLens100K, Election A5, Election A9, Election A17, Election A48, Election A81, the standard deviation of prediction accuracy is below 0.001. For datasets whose Intransitivity Ratio is relatively high, e.g., SF4-5000, DotA, Pokemon, the standard deviation of predictive test accuracy is reasonably higher at above 0.002.

A summary of CPU time for training the models in Chapter 3 and Chapter 4 is provided in Table 4.3. The CPU time of each method is reported based on the experiments on the SushiB dataset on a Macbook Pro with 2.9 GHz Dual-Core Intel Core i5 and 16 GB 1867 MHz DDR3 memory. Because the designs and implementations of each model and the corre-

Table 4.3: A summary of model parameters and the CPU time (in seconds) for SushiB

Method	Model Parameter(s)	CPU time for Training
Bradley-Terry	$\lambda \in \mathbb{R}^1$	0.52
Blade-Chest-Inner	$\mathbf{a}_{blade}, \mathbf{a}_{chest} \in \mathbb{R}^d$	8.70
Blade-Chest-Sigma	$\mathbf{a}_{blade}, \mathbf{a}_{chest} \in \mathbb{R}^d$	8.04
Generalized BC (Chapter 3)	$\mathbf{a}_{general} \in \mathbb{R}^{2d}, \Sigma, \Gamma \in \mathbb{R}^{d\times d}$	19.88
Neural BC	$\mathbf{a}_{blade}, \mathbf{a}_{chest} \in \mathbb{R}^d, w_{ij} \in \mathbb{R}^d$	29.10
Proposed Model (Chapter 4)	$w_{ij} \in \mathbb{R}^d$	28.59

sponding stochastic optimisation follow similar principles (e.g. cross-validation configuration, convergence criteria, gradient acceleration, etc.), we argue that the CPU time listed in Table 4.3 provides additional insights on how different the methods are. Bradley-Terry model has the simplest model parameter structure and reports the lowest CPU time for training. The proposed method in Chapter 3 has a moderate model complexity, because of the additional numerical calculation in the gradient estimation and during the inference, caused by the transitive matrics Σ and Γ. The highest CPU time is reported from the experiments on the models that are structured by deep neural networks, because updating the meta weights w_i, j in the neurons requires additional numerical operations like back propagating the errors from the output layer to the hidden layers. Nevertheless, the gain on predictive accuracy achieved by the deep models compensates well for the increase in CPU time cost.

4.6 Summary

In this chapter, we proposed a novel end-to-end intransitivity modeling technique for predicting pairwise comparisons and matchups. Different from the techniques presented in Chapter 3, a structured multi-layer neural network is used for the parameter estimation instead of logistic function. By conducting missing value inference experiments on synthetic data with controlled rank, we verified that the proposed end-to-end approach is superior to the shallow models for datasets that contain different intrinsic dimension. Through similar experimental setting in Chapter 3, we demonstrated that the proposed method improves the prediction accuracy in comparison to the Blade-Chest model and other related works. Moreover, we argue that the proposed framework is also open for further extensions to more expressive network structures with the recent advancements in deep neural networks, e.g., attention mechanism.

Chapter 5

Learning to Rank for Multi-step Ahead Time Series Forecasting

5.1 Introduction

Time-series forecasting is a fundamental problem associated with a wide range of engineering, financial, and social applications. The challenge arises from the complexity due to the time-variant property of time series and the inevitable diminishing utility of predictive models. Therefore, it is difficult to accurately predict the sequential values given the uncertainty of observations, especially in a multi-step ahead setting. Instead of forecasting the values of the time series, an ex-ante prediction of the relative order of values in the near future is sufficient in domains such as financial time series forecasting; i.e., the next 100 days can help make meaningful investment decisions.

The data veracity of time series refers to how accurate or trustworthy the collected sequential observations are, if a ground truth exists. For time series, the veracity issue is essentially associated with the stochastic nature of the sequential observations in the data acquisition process, in which setting a proper sampling frequency and annotating the cross-sectional alignment are more an art than a technical problem to solve. For practitioners, the veracity of

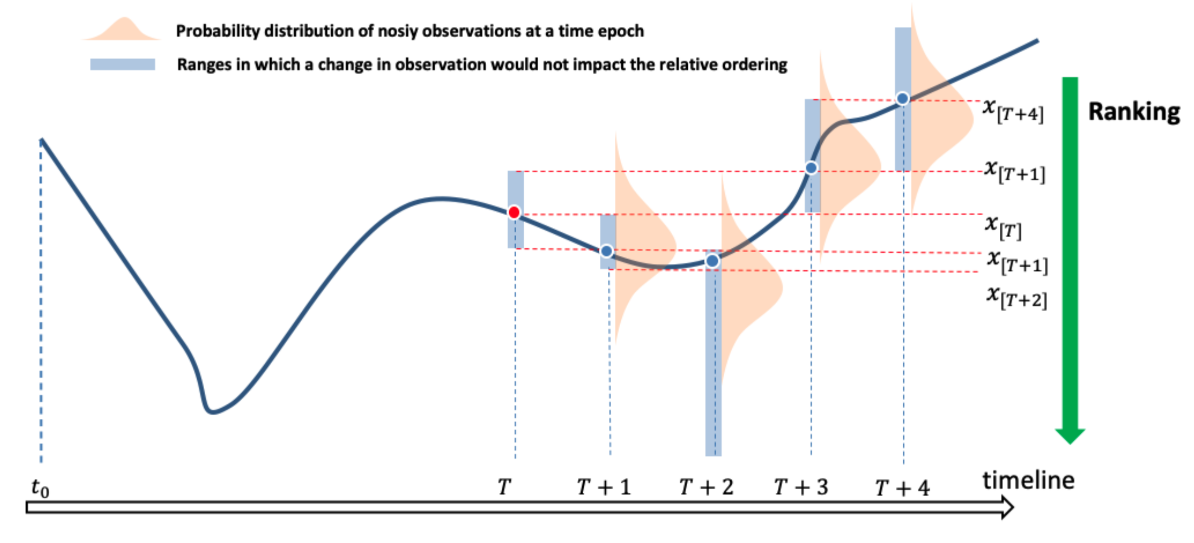

Probability distribution of nosiy observations at a time epoch
Ranges in which a change in observation would not impact the relative ordering
$x_{[T+4]}$
$x_{[T+1]}$
$x_{[T]}$
$x_{[T+1]}$
$x_{[T+2]}$
Ranking
t_0
T
$T+1$
$T+2$
$T+3$
$T+4$
timeline

Figure 5.1: Illustration of data veracity of an observed time series.

time series is therefore neglected, because the available time-series datasets only record one-way sequential observations. Consequently, it is difficult to take the veracity of the observed time series into consideration during the development of a forecasting algorithm. In contrast, if multiple samples with certain variation can be collected simultaneously at time epoch t, one can then utilize the samples to estimate a distribution or identify a proper stochastic process in order to quantify the magnitude of stochasticity. This would provide analytical insights on how responsive are the algorithms to the veracity of time series in a dataset. A traditional treatment to unveil such insights is block bootstrapping, a technique designed to quantify the impact of the stochasticity of sequential observations. However, the technique is an *ex-post* evaluation of the developed forecasting algorithms and does not tackle the challenge in robust forecasting under stochasticity in the front line.

Figure 5.1 illustrates how the veracity of time series may impact a time-series forecasting algorithm and thereby highlights the importance of evaluating the relative ordering of observations as an alternative and meaningful perspective to handle the veracity of time series. At each time epoch, the observed value follows an unknown distribution, however, the data acquisition process has only limited budget or capacity to simultaneously obtain enough samples from these distributions. For each of these distributions, training objectives that measures the deviation between observed value and the mean of the underlying distribution are subject to changes in the observation, even if the change is small. This is different when the relative ordering of sequential observations is considered because a subtle change of the sampled value from the distribution would make no impact on the relative ordering of the observations, as long as the it falls in a trustworthy zone, marked in blue.

In this chapter, we propose a dynamic prediction framework that makes it possible to make an ex-ante forecast of time series with a special focus on the relative ordering of the forecast within a forward-looking time horizon. Through the lens of the concordance index (CI), we compare the proposed method with conventional regression-based time-series forecasting methods, discriminative learning methods and hybrid methods. Moreover, we discuss the use of the proposed framework for different types of time series and under a variety of conditions. Extensive experimental results on financial time series across a majority of liquid asset classes show that the proposed framework outperforms the benchmark methods significantly.

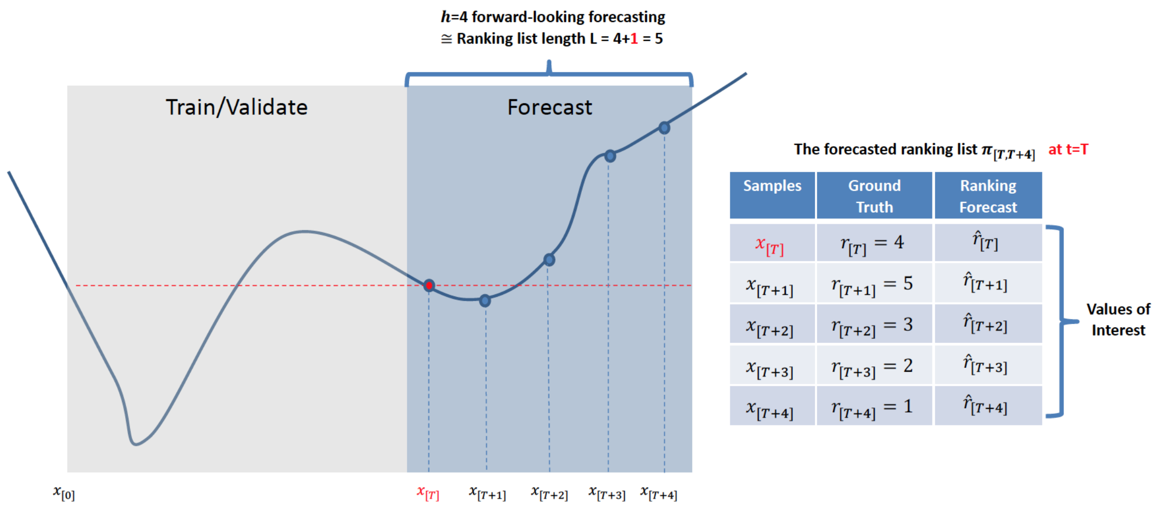
In science and engineering applications, time-series forecasting is applied to areas such as energy management [102], predictive maintenance [107], and anomaly detection [108]. Normally, it is evaluated based on the nowcasting performance, which reduces to certain evaluation metrics such as the tracking error. In the financial and social domains, the impact of time-series forecasting goes beyond nowcasting and it shifts its focus from the near future to the long-term horizon, bringing in other perspectives such as concordance, causality, in order to guide the decision makers to intervene appropriately. In this case, the use of a forecasted time series is prioritized over the conventional tracking error. Examples of this include empirical economics [113, 112], asset pricing [109, 95], business cycle analysis [110], monetary policy [111], and others that span all of the UN Sustainable Development goals, which address a blueprint for achieving a better and more sustainable future for all [98]. In addition to this discretion of appropriate utility function and evaluation metric, the length of the forward-looking horizon is an equally important aspect for such time-series forecasting task [120]. However, the majority of research work in time-series forecasting focuses on short-term forecasting, and often even on one-step ahead setting. For this reason, the potential variations in optimization objectives and evaluation metrics are not well explored beyond a predominant focus on tracking error. On the one hand, it is trivial to show the deterioration in forecast quality assuming a random walk prediction with no prior knowledge of what will happen in the next time epoch. On the other hand, the conviction that near-term nowcasting is accurate can provide meaningful support for long-term forecasting, especially in applications where the sequential dependency matters for multiple time epochs of interest. At the intersection of nowcasting and long-term forecasting, the predictability of time series is involved not only theoretically but also empirically [117]. Assuming that the forecasting task is ergodic, the predictability can be formulated as the ratio of the variance in the optimal prediction to the variance in the ground truth time series. This sheds light on feasibility issues in time-series forecasting.

To address the many challenges in time-series forecasting, a variety of time-series forecasting approaches have been developed to capture certain structural assumptions of time series. Traditional methods include the non-stationary model [101, 102], the moving average method [100], the auto-regressive model [99], the auto-regressive moving average model [149], the auto-regressive integrated moving average (ARIMA) model [100], the tree-based model [158], and the fuzzy time series models that consider different types of uncertainties, varying from the formulation in evidence theory [151] to the formulation in statistics or fuzzy logic and set [150]. Independently, machine learning and deep neural network approaches, which have been developed in the past few decades with a focus on discriminating between observations, have also been adopted in time-series analysis to tackle the forecasting problem [116, 118, 158, 159, 114, 92, 87, 153]. Overall, the above techniques provide sound founding elements for time-series analysis and forecasting, but the simplicity of the model results in its limited capability to deal with sophisticated situations. Moreover, because many of these techniques are derived independently in terms of notations and terminology, the alignment and synergy between these methods become extremely challenging [155]. As a result, the implementation of machine learning methods in the context of time-series forecasting oversimplifies the time-series forecasting problem by assuming i.i.d. samples and neglecting the sequential nature of the observed signals. Although the actual utility of the forecast in a one-step ahead setting varies by application, the corresponding evaluation metric is often monotonously inherited from that of regression-based methods. In such scenarios, the common determinant criterion is the tracking error calculated from the point-wise difference between the ground truth and the forecasted value. Among the most widely used performance metrics in this category [121], the symmetric mean absolute percentage error (SMAPE) and the mean absolute scaled error are frequently used in existing literature [148, 146].

However, in a multi-step ahead setting, tracking error is no longer the only aspect of interest in performance evaluation [115]. Depending on how the forecasted values are further utilized, other discriminative metrics, such as directional symmetry [115], trajectory affinity [122], relative orders and concordance [123], become equally important or even more important when evaluating performance. For instance, in situations where privileged information is available, local forecasting around the privileged time epoch is less vulnerable from the perspective of tracking error [103], whereas the relative disordering phenomenon remains considerable. Among others, the relative ordering and concordance of the forecast are unique and critical to problems where the structural insights of time series matter [145].

Recently, the advancement of time-series forecasting methods has been featured by the construction of hybrid methods and the use of alternative perspectives. The traditional formulation and evaluation can be extended to a multi-step ahead forecasting setting by introducing a structured output, e.g., multiple output and recursive output. Such forecasting strategies and the essential techniques for multi-step ahead settings have been comprehensively reviewed by [119, 152]. Moreover, in contrast to traditional approaches, many studies have begun to adopt novel perspectives for time-series forecasting. Hybrid methods have proposed to highlight the benefits of combining the traditional time-series forecasting models with alternative objectives or complementary techniques [156, 157]. Some examples include the use of complex networks [147, 146], the ordered weighted averaging aggregation operator [148] and deep-neural-networks-based approaches [154, 103, 153]. These hybrid methods bring together different objectives from a mathematical programming perspective and some even enhance the expression power of existing architectures by incorporating deep neural networks [133, 153]. As with the growth of model complexity and the number of hyperparameters, the generalization of a model's predictive power in a multi-step forecasting setting becomes a greater challenge. Rather than contributing to this increasing sophistication, we argue that alternative objectives of certain time-series forecasting models, which are expected to deliver a higher explanation power by leveraging a limited number of model parameters, are critical to the time-series forecasting research. Specifically, in this study, we investigate the relative ordering objective, which has not yet been thoroughly formulated and explored.

In the financial domain, the relative ordering of the market values of asset prices at different time epochs is an essential component, because it is associated with theoretical and practical issues, e.g., mispricing, arbitraging, and market inefficiency. On the hunt for excessive investment returns, asset managers and hedge funds can leverage a variety of financial instruments, including futures, swaps, and options to monetize investment ideas and to optimize the investment performance given the investment ideas originated from such relative ordering. However, despite the wide adoption of stochastic mathematics and regression-based techniques in time-series analysis, limited efforts have been made to explore the relative orderings of time series within a specified horizon. Therefore, developing a method to forecast the relative ordering of the observations in time series is fundamental to time-series forecasting and critical in many financial applications [124, 91]. To an extreme degree, census data or variables in macroeconomics are released at a lower frequency in contrast to the data available at the



The forecasted ranking list $\pi_{[T,T+4]}$ at t=T

Samples	Ground Truth	Ranking Forecast
$x_{[T]}$	$r_{[T]} = 4$	$\hat{r}_{[T]}$
$x_{[T+1]}$	$r_{[T+1]} = 5$	$\hat{r}_{[T+1]}$
$x_{[T+2]}$	$r_{[T+2]} = 3$	$\hat{r}_{[T+2]}$
$x_{[T+3]}$	$r_{[T+3]} = 2$	$\hat{r}_{[T+3]}$
$x_{[T+4]}$	$r_{[T+4]} = 1$	$\hat{r}_{[T+4]}$

Figure 5.2: Illustration of the multi-step time-series forecasting scheme. There are in total five values of interest including the $\hat{r}_{[t]}$, whose ground truth of $x_{[t]}$ is known and highlighted in red color. ©IEEE, 2021

exchange markets. This results in a gap between the tremendous number of model parameters and the small number of available observations [160], which further induces difficulty in model generalization. As an addition to the traditional econometric approaches, alternative methods [166, 165, 163] have been proposed to achieve robust parameter estimation, where vector auto-regressive models [160, 161, 163] suffer from an out-of-sample prediction performance in the mid-long horizon. The intrinsic problem related to such application is that directional guidance and timing become more important issues than tracking error [164, 167]; therefore, alternative techniques that can leverage alternative utility functions of the forecast values should be highlighted [162].

To tackle these challenges, we propose a multi-step ahead forecasting framework that is capable of forecasting relative orders at multiple time epochs within a forward-looking horizon. The framework has three key components, including pairwise discriminative learning, local learning (LL) of privileged information, and dynamic multi-step ahead prediction with ex ante information. First, the pairwise discriminative learning module follows the learning-to-rank principles that directly optimise ranking objective rather than tracking error and output a ranking list instead of an array of forecasted values. Second, a LL algorithm is proposed to infer the values with the concept of neighboring or auxiliary samples, so that the optimised ranking list is interpretable and comparable to the conventional forecasts. Finally, all the above proposed elements are integrated into a dynamic multi-step ahead prediction scheme iteratively, aiming to boost the overall predictive performance. By bridging the best of two orthogonal evaluation metrics, i.e. relative ordering and tracking error, this scheme delivers an overlaying synergy of all the proposed elements underneath.

5.2 Preliminaries

In this section, we describe the problem setting and the notations. As depicted in Section 5.1, one major challenge in multi-step ahead forecasting is the domain-specific utility of the forecasted values. The other challenge is the diminishing utility of the trained model when the forecast horizon is expanded from one-step ahead to multi-step ahead.

To tackle the first challenge, we employ the learning-to-rank technique, which is considered as a fundamental method for ranking estimation, recommender systems, and data science [3]. Although being unexplored in time-series forecasting, the objective of learning-to-rank technique is to minimize the mismatch between the ground-truth ranking list and the estimated ranking list, which is in practice orthogonal to the conventional objective of minimizing the point-to-point euclidean error. In recent years, a variety of machine learning approaches to ranking estimation have been discovered via techniques such as Bayesian modeling [125], generalization of the Bradley-Terry model [72], the structured support vector machine (SVM) [49], optimal transport [126], the tree-based models [159], and fuzzy logic [148, 147, 146], etc. Different from the others, ranking SVM directly models and optimises the ranking loss on the data, which is essential to the ranking estimation problem. Although we only consider the linear ranking SVM in this study, the formulation is flexible for further extension to tackle the non-linear features via kernel tricks with a minor revision of the framework; therefore, ranking SVM is favourable among others, especially in applications where the availability of high quality features is limited. Overall, formulating time-series forecasting in a ranking setting remains an open and challenging problem. In this study, we leverage the ranking SVM because it tackles the ranking estimation problem with an end-to-end objective and with a favourable flexibility. In the following parts of this section, we establish the notations and define the problem setting, which are innovative in themselves.

To tackle the second challenge, we devise an iterative prediction scheme that follows the dynamic forecasting strategy in [119] in order to make informative inference. This scheme is motivated by LL techniques that bridge the two orthogonal worlds of tracking error and relative ordering. Although the details will be discussed in Section 5.3.4, the key notations and preliminaries will be described in this section.

5.2.1 Problem setting

We consider multi-step time-series prediction. Assume $\mathbf{X}_{[0:N]}$ is the ground truth time series with a span of $[0 : N]$. At time t, the observation of the time series is given as $X_{[t]} = (x_{[t]}, r_{[t]}) \in \mathbf{X}$, where $x_{[t]} \in \mathbb{R}^p$ is the observation of the time series and $r_{[t]} \in [0 : N]$ is the associated rank of the observation within the whole span of the time series during $[0 : N]$. We define $\mathcal{R} : \mathbb{R}^p \to [n]$ as an invertible ranking function that transforms $x_{[t]}$ to $\hat{r}_{[t]}$. The corresponding inverse function is denoted as $\mathcal{R}^{-1} : [n] \to \mathbb{R}^p$ and transforms $r_{[t]}$ to $\hat{x}_{[t]}$. The transform function is formulated as

$$\hat{r}_{[t]} = \mathcal{R}(\hat{x}_{[t]}), \tag{5.1}$$

$$\hat{x}_{[t]} = \mathcal{R}^{-1}(\hat{r}_{[t]}, x_{[0:t]}). \tag{5.2}$$

The goal is to make an informed forecast of a ranking list $\pi_{[t]}^{(h)} := \hat{r}_{[t+1:t+h]}$, which is a ranking list of length h that expresses the relative relationships of the observations that are obtained from the multi-step forecast $\hat{X}_{[t+1:t+h]} = (\hat{x}, \hat{r})_{[t+1:t+h]}$ at time $t = T$. Given all the historical observations $(X_{[0]}, X_{[1]}, ..., X_{[T]})$ at $t = T$, we aim to forecast the relative relationships in the time series in a forward-looking horizon of h during $[T + 1 : T + h]$, where the ground truth of the forecast $r_{[T+1:T+h]}$ is a subset of $\mathbf{X}_{[0:N]}$. Without loss of generality, by setting $p = 1$, we address the problem of forecasting time series with only one dimension. However, to implement all the competing methods and the proposed methods in a fair and qualitative manner, we construct multidimensional features based on generic feature engineering techniques for time series, especially time series in the financial domain.

In this study, we use $\pi^{(h)}_{[t]} \in [0 : h]$ to denote the h-step ahead forecast of relative relationships at time t. An auxiliary set $\tilde{x}_{[t-q:t]} = x_{[t-q:t]}$ representing the q auxiliary anchors is deployed in LL to boost the performance of the forecast. The simplest case is $q = 0$, indicating no privileged information other than the observation at time t can be utilized during the inference phase. However, in common cases, it is natural to assume that we have access to and can reuse all the historical observations up to time t. In the extreme scenario where $q = t$, the auxiliary set is identical to the training set and is denoted by $\tilde{x}_{[t-q:t]} = x_{[0:t]}$.

Figure 5.2 illustrates this relative-value-focused multi-step time-series forecasting scheme with an example that uses $h = 4$ and $q = 0$. In this extreme case, $x_{[t]}$ is the only available privileged information in inference, and therefore it becomes difficult to infer $\hat{x}_{[t+1:t+h]}$, regardless of whether the ranking estimation is perfectly aligned with the ground truth ranking or not within the test horizon. In a nutshell, only $x_{[t]}$ can be exploited for a hint as to whether the observations within the test horizon will be higher or lower than $x_{[t]}$. In cases where the relative ordering is not the only metric of interest, the learning-to-rank model requires deliberate revision before it can provide meaningful guidance.

5.2.2 Feature extraction

Among the various techniques that have been proposed to extract informative features from time series, we adopt two fundamental suites of indicators for financial time series, i.e., level indicators and momentum indicators [91].

Level indicators consist of the historical prices denoted by $F_{price}(t,i)$ and the moving averages of the historical prices [99] denoted by $F_{ma}(t,i)$, for $i \in \{5, 10, 20, 60, 120, 250, 500, 1000\}$, where

$$\begin{aligned} F_{price}(t, i; x_{[0:t]}) &= x_{[t-i]} && (5.3) \\ F_{ma}(t, i; x_{[0:t]}) &= \frac{\sum\limits_{j=1}^{i} x_{[t-i]}}{i} && (5.4) \end{aligned}$$

Momentum indicators include the moving average convergence divergence (MACD) [129] and rolling returns over the past $\{5, 10, 20, 60, 120, 250, 500, 1000\}$ days. In consideration of the fact that volatility is a critical facet of financial time series and the fact that the volatility scaling technique plays a significant role in the construction of investment strategies [128], we adopt the risk-adjusted momentum features $F_{rollingReturn}(t,i)$ by

$$F_{rollingReturn}(t, i; x_{[0:t]}) = \frac{x_{[t]} - x_{[t-i]}}{std(\dot{x}_{[t-i:t]})} \tag{5.5}$$

where $std(\dot{x}_{[t-i:t]})$ is the standard deviation of the daily price change $\dot{x}$ during the period $[t-i, t]$.

With the feature extraction defined as above, we construct multidimensional features for each observation $x_{[t]}$ that is obtained from the environment at time t. Finally, the level indicators and momentum indicators are combined by a concatenation denoted by

$$F_{all}(t, i; x_{[0:t]}) = [F_{price}, F_{ma}, F_{rollingReturn}] \tag{5.6}$$

where $i \in \{5, 10, 20, 60, 120, 250, 500, 1000\}$. Overall, 17 features are employed this study for the proposed framework.

5.3 Proposed Methods

5.3.1 Learning to rank for time-series forecasting

We formulate the multi-step time-series forecasting task as a multi-step learning-to-rank process that incorporates a learning phase and an inference phase. In the inference phase, a ranking model takes the historically observed objects and their associated feature matrices $\mathbf{X}_{[0:t]}$ as an input and outputs the inferred scoring of the observed objects $\hat{\pi}_{[0:t]}$.

$$\hat{\pi}_{[t:t+h]} = f(\mathbf{X}_{[0:t]}, \mathbf{W}; w) \tag{5.7}$$

The overall learning objective of the ranking model is to discriminate the observed objects from a relative relationship perspective so that the inferred ranking list $\hat{\pi}_{[t:t+h]}$ is close enough to the ground truth $\pi_{[t:t+h]}$. Assuming the observations being forecasted, $\mathbf{X}_{[0:t]}$, by the ranking model follow the same distribution of historical observations, the learning objective in Equation (5.7) is approached by devising a pairwise ranking model based on the observed time series by time t. Given a rank order $r_{[i]} \in \pi_{[0:t]}$ where $i \in [0 : t]$, a scoring function $\phi(\mathbf{X}_{[i]})$ is devised to measure the relative ordering of the objectives.

$$\begin{aligned} \phi(i) &= \phi(\mathbf{X}_{[i]}; w) && (5.8) \\ &= \sum_j w^j \mathbf{X}^j_{[i]} && (5.9) \end{aligned}$$

An array of such scoring $\phi(\cdot)$ can be trivially converted to a corresponding ranking list by using Equation (5.1). In fact, it is identical to the primal model output $\hat{\pi}_{[t,t+h]}$ from the ranking perspective and is used as an input in the calculation of the ranking performance metric, which we will introduce later in Section V.

Assume that $\mathbf{W} \in \mathbb{R}^{t,t}$ is a match-up matrix indicating the soft constraints that can be incorporated into the learning part of the framework, where an entry $W_{i,j}$ denotes the importance weight of each pairwise comparison (i, j) in the ranking list. Given the ground truth $\pi_{[0,t]}$, each entry in the match-up matrix $W_{i,j}$ is defined as follows:

$$W_{i,j} = \begin{cases} +1 & \text{if } r_{[i]} \succeq r_{[j]} \\ -1 & \text{if } r_{[i]} \prec r_{[j]} \\ 0 & \text{pair is not considered} \end{cases}$$

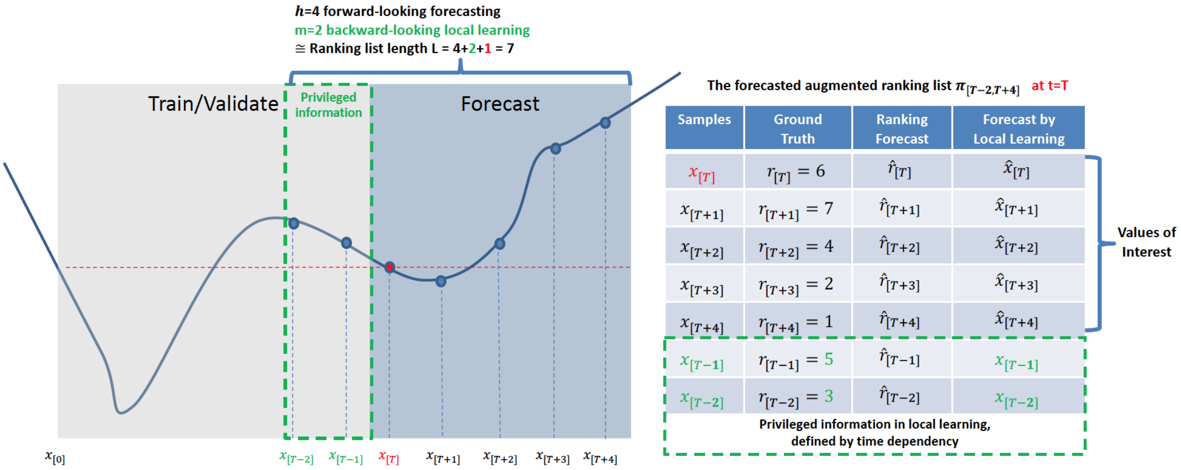
where $i, j \in [0 : t]$. The ranking loss between the ground truth π and the inferred ranking list $\hat{\pi}$ is expressed as

$$\Delta(i, j) = W_{i,j}(\phi(\mathbf{X}_{[i]}; w) - \phi(\mathbf{X}_{[j]}; w)) \tag{5.10}$$

where $\phi(\mathbf{X}_{[i]}; w)$ is the scoring function defined by Equation (5.8) for the observation at time i.

Note that the feature for the observation at t is often a function of the observation by that time; however, the values are not available between t and $t + h$ in multi-step ahead setting. Therefore, the construction of pseudo features is necessary for forecasting the future observation at $t + h$. As a naive solution, we adopt the latest available observation for the ex-ante observations at each future time epoch $t + h$, formulated as

$$\tilde{x}_{[t+j]} = x_{[t]} \tag{5.11}$$



The forecasted augmented ranking list $\pi_{[T-2,T+4]}$ at t=T

Samples	Ground Truth	Ranking Forecast	Forecast by Local Learning
$x_{[T]}$	$r_{[T]} = 6$	$\hat{r}_{[T]}$	$\hat{x}_{[T]}$
$x_{[T+1]}$	$r_{[T+1]} = 7$	$\hat{r}_{[T+1]}$	$\hat{x}_{[T+1]}$
$x_{[T+2]}$	$r_{[T+2]} = 4$	$\hat{r}_{[T+2]}$	$\hat{x}_{[T+2]}$
$x_{[T+3]}$	$r_{[T+3]} = 2$	$\hat{r}_{[T+3]}$	$\hat{x}_{[T+3]}$
$x_{[T+4]}$	$r_{[T+4]} = 1$	$\hat{r}_{[T+4]}$	$\hat{x}_{[T+4]}$
$x_{[T-1]}$	$r_{[T-1]} = 5$	$\hat{r}_{[T-1]}$	$x_{[T-1]}$
$x_{[T-2]}$	$r_{[T-2]} = 3$	$\hat{r}_{[T-2]}$	$x_{[T-2]}$

Privileged information in local learning, defined by time dependency

Figure 5.3: Illustration of the local learning that is embedded into the dynamic prediction scheme for multi-step time-series forecasting scheme. ©IEEE, 2021

for all $j \in [0 : h]$. Therefore, the features $\tilde{\mathbf{X}}_{[t+h]}$ are calculated by calling the feature construction function with the above pseudo estimates by

$$\tilde{\mathbf{X}}_{[t+h]} = F_{all}(t+h; x_{[0:t]}, \tilde{x}_{[t:t+h]}) \tag{5.12}$$

The overall loss function is written as

$$L(\mathbf{X}, \mathbf{W}; w) = \frac{1}{|\mathcal{P}||\mathcal{N}|} \sum_{x \in \mathcal{P}} \sum_{y \in \mathcal{N}} d(\Delta(i,j)) \tag{5.13}$$

where

$$d(\Delta(i,j)) = \max(0, 1 - \Delta(i,j)) \tag{5.14}$$

is the Hinge loss [130]. In Equation (5.13), $\mathcal{P}$ and $\mathcal{N}$ denote the upper-ranked object subset and the lower-ranked object subset, respectively. Without loss of generality, one can extend the definition of $\mathcal{P}$ and $\mathcal{N}$ to the full list, which is also referred to as the list-wise ranking operation [49].

In our proposed framework, we restrict ourselves to solving the classic RankSVM optimization problem [130, 173, 49]. The overall optimization objective is expressed as

$$\min_{w} \frac{\lambda}{2}||w||^2 + L(\mathbf{X}, \mathbf{W}; w) \tag{5.15}$$

where λ is the hyperparameter regularizing the learning objective in SVM via the soft margin embedded in the first term of the optimization objective in Equation (5.15).

5.3.2 Optimization

Solving for w^* in Equation (5.15) is a numerical optimization problem. Generic solutions to such a problem include the stochastic subgradient descent method in the Support Vector Machine optimization [168, 71]. Specifically, we adopt the stochastic subgradient descent method for solving the L2-regularized L1-loss SVM implemented in the scikit-learn library [131] as the solver for our proposed pairwise learning-to-rank method. The subgradient update is given as

$$w \leftarrow w - \eta \nabla_w (W_{i,j} w^T (\mathbf{X}_{[i]} - \mathbf{X}_{[j]})) \tag{5.16}$$

where $\eta > 0$ is the learning rate and is chosen as $\eta(t) = \frac{1}{\lambda(t+t_0)}$. t is the epoch of gradient update and t_0 is chosen by the heuristic proposed in [169]. For a cross-temporal pair (i, j) sampled from the training set, the corresponding subgradient is estimated by

$$\nabla_w(i,j) = \begin{cases} \lambda w & \text{if } W_{i,j}(\phi(i) - \phi(j)) > 1, \\ \lambda w - W_{i,j}(\mathbf{X}_{[i]} - \mathbf{X}_{[j]}) & \text{otherwise} \end{cases}$$

The selection of regularization term λ was done through a set of predefined hyperparameters $\{10^{-4}, 10^{-2}, ..., 10^{4}\}$. Empirically, we would like to emphasize that the pairwise ranking

Algorithm 1 Pairwise Ranking - Learning to rank for multi-step ahead time-series forecasting

```
Input: Forward-looking horizon $h$
Data: Observed time series $x_{[0:t]}$
initialization
for $i \in [0:t]$ do
    $\mathbf{X}_{[i]} \leftarrow F_{all}(i; x_{[0:t]})$;
    $r_{[i]} \leftarrow \mathcal{R}(x_{[i]})$;
end
$\hat{w}^* \leftarrow$ Optimize Equation (5.15)
for $j \in [0:h]$ do
    $\hat{\phi}(t+j) \leftarrow f(\mathbf{X}_{[t+j]}; \hat{w}^*)$;
    $\hat{r}_{[t+j]} \leftarrow \mathcal{R}(\hat{\phi}(t+j))$;
end
Output: Forecast of $\hat{\pi}_{[t:t+h]}$
```

approach faces certain issues like scalability when the volume of training data grows and computational convergence when the learning rate is not chosen properly. However, with a fine-tuning of the key hyperparameters, e.g. the learning rate η and the λ associated with the regularization term as we suggested, these issues can be mitigated.

Algorithm 1 describes the proposed learning-to-rank framework for multi-step ahead time-series forecasting. The input of the algorithm is the forward-looking horizon. After initialization of the model parameters including w, λ, and the forward-looking horizon h, the pairwise learning-to-rank model is trained until the convergence condition is met. This training phase is followed by a standalone inference phase where the forecast of relative ordering $\hat{r}_{[t:t+h]}$ is inferred.

Note that the $\hat{r}_{[t:t+h]}$ estimated by Algorithm 1 is insufficient for an estimation of the tracking error, which is normally adopted by competing methods and related works on multi-step ahead time-series forecasting. To fill in this gap between the orthogonal evaluation metrics, we further develop a LL technique in Section 5.3.4 that is able to obtain $\hat{x}_{[t+1:t+h]}$ and integrate the LL module into an overlaying dynamic forecasting scheme that operates iteratively between

Algorithm 2 Local learning of $\hat{x}_{[t+1:t+h]}$

```
Input: Backward-looking horizon $m$
Data: Observed time series $x_{[0:t]}$
initialization
$\hat{r}_{[t:t+h]} \leftarrow$ Algorithm 1
$\hat{r}_{[t-m:t]} \leftarrow$ Equation (5.7)
$\hat{r}_{[t-m:t+h]} \leftarrow$ Merge $\hat{r}_{[t-m:t]}, \hat{r}_{[t:t+h]}$
for each $j \in [t+1:t+h]$ do
    $j^+ \leftarrow \underset{p\in[t-m:t]}{\text{argmin}} \hat{r}_{[p]} - \hat{r}_{[j]}$ for $\hat{r}_{[p]} > \hat{r}_{[j]}$
    $j^- \leftarrow \underset{p\in[t-m:t]}{\text{argmax}} \hat{r}_{[p]} - \hat{r}_{[j]}$ for $\hat{r}_{[p]} < \hat{r}_{[j]}$  $\hat{x}_{[j]} = \frac{1}{2}(x_{[j^+]} + x_{[j^-]})$
end
Output: Forecast of $\hat{x}_{[t+1:t+h]}$
```

learning and inference to lower the tracking error while optimizing the discriminative objective in Equation (5.15).

5.3.3 Local learning

To improve the quality of the multi-step ahead forecast, we apply LL techniques during the approximate inference of $\hat{x}_{[t+1:t+h]}$. In Algorithm 1, the output of the algorithm is the relative ordering of objects, which reveals only the relative relationship between future observations and the observation at time t. We devise a local learning(LL) procedure to improve the inference of $\hat{x}_{[t+1:t+h]}$, which can be used in turn to improve the quality of features by letting

$$\tilde{x}_{[t+j]} = \hat{x}_{[t+j]} \tag{5.17}$$

where $j \in [1:h]$ and $\tilde{x}_{[t]}$ represents the ex-ante expected value of the time series at time t.

Figure 5.3 illustrates the proposed LL scheme. In this study, we define the privileged information available for local learning from the perspective of time dependency [120]. At time T, the most recent $m \leq T$ samples are employed for local learning and we update the point estimate each time when the local learning procedure is called.

To further boost the forecasting performance based on the informative ex-ante prediction obtained from local learning, we devise a dynamic prediction as described in Algorithm 3.

Local learning is an important technique in machine learning that allows the model to efficiently incorporate certain dependency structures, such as neighborhood dependency [132], time dependency [120], and spatial-temporal dependency [133]. Given an output $\hat{r}_{[t:t+h]}$ from Algorithm 1, auxiliary information can be added as anchors into the well-trained ranking model, so that the inference of $\hat{x}_{[t+1:t+h]}$ can be formulated as a LL procedure as described in Algorithm 2.

Assuming m neighbors are available for the LL, we rewrite Equation (5.1) in array form and incorporate the local neighbors as anchors by

$$\hat{x}'_{[t+1:t+h]} = \tilde{\mathcal{R}}^{-1}(\hat{r}_{[t-m:t+h]}, x_{[t-m:t]}) \tag{5.18}$$

The effectiveness of such an ex-ante forecast can be verified via performance metrics such as the concordance index (CI). Despite of the orthogonality between CI and tracking error, this promising result sheds light on a potential improvement in terms of the ex-ante tracking error; therefore, we propose an iterative dynamic forecasting scheme to capture this signal and combine the best of LL framework with the proposed learning-to-rank framework for time-series forecasting.

In Figure 5.3, we illustrate a toy example of local learning, where we indent to forecast five values of interest in the forward-looking horizon, with two privileged samples in the training and validation set highlighted in green color. In this example, the total length of the ranking list is seven, among which five are identical to the values of interest illustrated in Figure 5.2. What makes a difference is that we have partial access to their ground truth; therefore, a performance improvement in the inference phase can be expected.

Algorithm 3 DynaPairwise Ranking - Dynamic prediction scheme that improves learning to rank for time-series forecasting and local learning iteratively

```
Input: Maximum number of iteration $l_{max}$
Data: Observed time series $x_{[0:t]}$
initialization
 $\hat{r}_{[t:t+h]} \leftarrow$ Algorithm 1
while $l \leq l_{max}$ do
    $\hat{x}_{[t+1:t+h]} \leftarrow$ Algorithm 2
    $\hat{r}_{[t:t+h]} \leftarrow$ Algorithm 1 $l$++
end
Output: Forecast of $\hat{x}_{[t+1:t+h]}$, $\hat{r}_{[t+1:t+h]}$
```

5.3.4 Dynamic prediction

The previous section showed that multi-step ahead forecasting can be cast in a conventional supervised learning-to-rank framework by employing certain inference techniques such as LL. In this section, we extend the proposed framework in Algorithm 2 and further propose an iterative dynamic prediction scheme that improves the performance of multi-step ahead forecasting. The proposed scheme follows the recursive strategy for multi-step ahead time-series forecasting [120]. Such recursive strategies for time dependency are widely deployed implicitly or explicitly in structured output prediction models such as conditional random fields [135] and recursive neural networks [134].

Overall, the proposed iterative dynamic scheme involves three steps as described in Algorithm 3. The key components are organized in a nested structure. In the first step, Algorithm 1 is called to give an initial ex-ante forecast of the relative ordering $\hat{r}_{[t+1:t+h]}$. Next, the LL procedure depicted in Algorithm 2 is called to provide an initial ex-ante forecast of the time series $\hat{x}'_{[t+1:t+h]}$. Because the features are constructed by a function of such an ex-ante estimate of the time series in Equation (5.6), Algorithm 1 is called again in step 3 to brush up the features by using these ex-ante estimations. Such an iterative procedure can be terminated either when a convergence criterion is met or when the maximum number of iterations l_{max} is reached. In

Table 5.1: Hyperparameters deployed in the benchmark scheme and the proposed method. ©IEEE, 2021

Parameter	**Setting**	**Description**
N_{train}	500	size of the training and validation data
N_{roll}	{50,100,200}	number of time epochs between model retraining
m	{0,100,500}	length of the backward-looking horizon in local learning
h	{50,100,200}	length of the forward-looking forecast horizon
l_{max}	1	number of iterations in the dynamic prediction scheme
λ	$\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$	strength of the regularization term that controls the soft margin
μ	refer to Table 5.2	calibration factor that reflects domain knowledge
η	refer to Equation (5.16)	learning rate of the gradient descent method

practice, we set $l_{max} = 1$ and apply the dynamic prediction procedure on a rolling basis with a specified interval that equals to the forward-looking horizon h.

5.4 Experiments

In this section, we describe the evaluation metrics, the characteristics of datasets and the benchmark procedure that is executed for all competing and proposed methods. In the experimental results, particularly in Table 5.4 and Table 5.5, we refer to Pairwise Ranking and DynaPairwise Ranking as the proposed Algorithm 1 and the proposed Algorithm 3, respectively.

5.4.1 Evaluation metrics

In line with the previous studies, three evaluation metrics, i.e., SMAPE [127], RMSE [121], and the CI [141, 142] are used to evaluate the performance of competing models including the theta model [136, 137], the regression-based model [138], the ARIMA model [100], the DeepAR model [153], the random forest model [158], the LightGBM model [159], the logistic regression model [139, 140], and the ordinal regression method [143].

Table 5.2: An overview of financial time series dataset. The dataset has two categories, i.e., growth indices and mean-reverting indices, addressing the economic properties of the underlying asset classes. In the upper part of the table, equity indices and fixed income indices are categorized as growth indices. In the lower part of the table, currency exchange indices are categorized as mean-reverting indices. ©IEEE, 2021

Index	**Asset Class**	**Ticker**	**Description**	**Experiment Period**	**Avg. Rolling Return (std.)**
Growth Indices					
S&P 500	Equity index	SPX	S&P 500 price return index	12.31.1999 to 12.31.2019	$5.19\%(16.11\%)$
Russel 2000	Equity index	RUT	Russell 2000 index	12.31.1999 to 12.31.2019	$7.27\%(19.21\%)$
DAX	Equity index	DAX	German stock total return index	12.31.1999 to 12.31.2019	$5.25\%(21.96\%)$
TOPIX	Equity index	TPX	Tokyo Stock Exchange price index	12.31.1999 to 12.31.2019	$2.58\%(22.34\%)$
Bund 10Y	Fixed income index	CBKIG0FT	German 10Y Bund futures index	12.31.1999 to 12.31.2019	$4.60\%(4.82\%)$
Gilt 10Y	Fixed income index	CBKIK0FT	UK 10Y Gilt futures index	12.31.1999 to 12.31.2019	$3.60\%(5.45\%)$
Treasury 10Y	Fixed income index	CBKIU0FT	US 10Y Treasury futures index	12.31.1999 to 12.31.2019	$3.82\%(5.13\%)$
Mean-reverting Indices					
EURUSD	Currency	EURUSDCR	EURUSD carry return index	12.31.1989 to 12.31.2019	$0.29\%(10.80\%)$
GBPUSD	Currency	GBPUSDCR	GBPUSD carry return index	12.31.1989 to 12.31.2019	$1.58\%(9.29\%)$
CHFUSD	Currency	CHFUSDCR	CHFUSD carry return index	12.31.1989 to 12.31.2019	$1.13\%(10.81\%)$
JPYUSD	Currency	JPYUSDCR	JPYUSD carry return index	12.31.1989 to 12.31.2019	$-0.72\%(11.05\%)$
CADUSD	Currency	CADUSDCR	CADUSD carry return index	12.31.1989 to 12.31.2019	$0.49\%(7.85\%)$
AUDUSD	Currency	AUDUSDCR	AUDUSD carry return index	12.31.1989 to 12.31.2019	$2.89\%(13.10\%)$
NZDUSD	Currency	NZDUSDCR	NZDUSD carry return index	12.31.1989 to 12.31.2019	$4.30\%(13.43\%)$
NOKUSD	Currency	NOKUSDCR	NOKUSD carry return index	12.31.1989 to 12.31.2019	$1.44\%(11.89\%)$
SEKUSD	Currency	SEKUSDCR	SEKUSD carry return index	12.31.1989 to 12.31.2019	$0.79\%(12.76\%)$

Mathematically, the metrics are calculated as follows:

$$\text{SMAPE} = \frac{1}{n}\sum_{t=1}^{n}\frac{\left|\hat{x}_{[t]} - x_{[t]}\right|}{\left|\hat{x}_{[t]}\right| + \left|x_{[t]}\right|} \times 100\% \tag{5.19}$$

$$\text{RMSE} = \frac{1}{n}\sqrt{\sum_{t=1}^{n}\left|\hat{x}_{[t]} - x_{[t]}\right|^{2}} \tag{5.20}$$

$$\text{CI} = \frac{2}{|\mathcal{E}|}\sum_{(i,j)\in\mathcal{E}} \mathbb{1}(\hat{r}_{[i]} \succeq \hat{r}_{[j]}) \tag{5.21}$$

where $\mathcal{E} := \{(i,j); x_{[i]} \geq x_{[j]}, i < j \in [1:n]\}$ is the set of event of interest (EOI), which is observed in the ground truth time series. $\mathbb{1}(\hat{r}_{[t]} \geq \hat{r}_{[t]})$ is an indicator function that outputs 1 when the condition $\hat{r}_{[t]} \geq \hat{r}_{[t]}$ is satisfied. At time i, we focus on the concordance between the relative ordering of the observations and the relative ordering of the model forecast. The CI is a generalization of the area under the ROC curve to the regression problem, and therefore it can be calculated trivially by transforming the continuous outputs into a ranking list using Equation

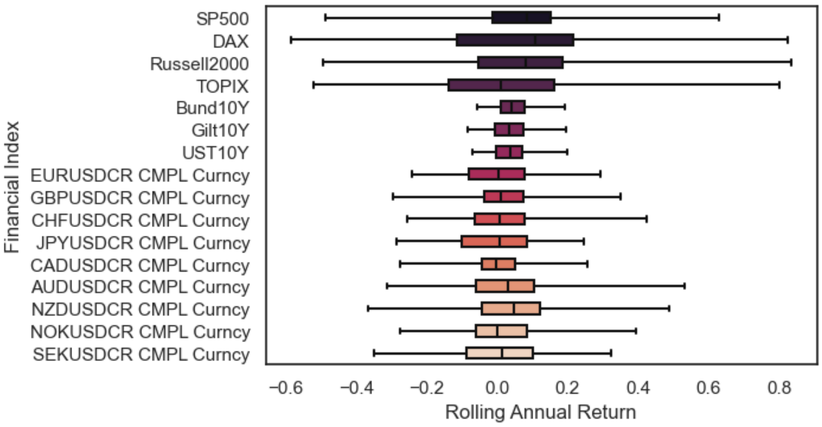


Figure 5.4: Two categories of financial indices, i.e., growth indices (in blue) and mean-reverting indices (in red). ©IEEE, 2021

(5.1). In our definition, over a specific time horizon n, the CI is 1 if all the upward movements of the time series along the timeline are successfully captured by the forecast and 0 if all the upward movements of the time series are forecasted as downward movements. Assuming the time series is a random walk with no expectations regarding its upside and downside, CI $= 0.5$ indicates that the performance is as good as a random predictor. However, given some domain knowledge of the time series, the outlook for certain directional move of the time series may shift from neutral to up or down in expectation. In such cases, the output of the models can be calibrated toward the predefined expectation by

$$\hat{x}'_{[t_0]}(t) = \hat{x}_{[t_0]} e^{\mu(t-t_0)} \tag{5.22}$$

where μ is a positive scalar when the outlook for the time-series observation from time t_0 is more on the upside and is negative when the outlook is more on the downside.

Note that for discriminative models, including logistic regression, ordinal regression, and the proposed pairwise ranking models, only $\hat{r}_{[t:t+h]}$ is forecasted. Therefore, the tracking errors can not be calculated and reported in the comparison with the competing methods. An exception is the proposed dynamic prediction scheme assembled with LL, for which the output of the discriminative model can be interpreted as scalar values $\hat{x}_{[t:t+h]}$ given sufficient privileged information. For non-discriminative models, including the theta model and the regression-based models, both $\hat{r}_{[t:t+h]}$ and $\hat{x}_{[t:t+h]}$ are forecasted. All three evaluation metrics are calculated and reported in Section 5.4.

5.4.2 Datasets

We use historical future contract and index data obtained from the Bloomberg Terminal[1]. The dataset is retrieved on October 31, 2020, and consists of major liquid asset classes including equity indices, fixed income indices, and currency indices[2]. An overview of the dataset is provided in Table 5.2. For the equity and fixed income indices, the observations are either price return or total return, which are tradable in the market via exchange-traded funds or future contracts at low transaction costs. The time horizon of the time-series data ranges from December 31, 1999, to December 31, 2019, on a daily basis. For currency indices, the time horizon is from December 31, 1989, to December 31, 2019, on a daily basis. In the last column of Table 5.2, the historical return statistics are presented to characterize the asset classes. In the last column of Table 5.2, we summarize the average annualized rolling return and the standard deviation of the annualized rolling return over the entire observation period. Because the observation period contains at least one economic and market cycle, the return and standard deviation statistics reflect the expected growth of an asset class in mid-long term. Depending on the level of growth expectation, we categorize the 16 indices into two groups. On the one hand, equity and fixed income indices are categorized as growth indices, because they generally record positive asset pricing results based on historical performance. This is in line with the economic outlook of these asset classes in the long run. On the other hand, currency indices are categorized as mean-reverting indices, because the annualized returns of the currency indices

[1] https://www.bloomberg.com/professional/

[2] Financial indices source data can also be obtained from other data vendors, e.g., https://tradingeconomics.com/

Table 5.3: A summary of averaged CPU time (in seconds) for model training with cross-validation.

$N_{\textbf{train}} = 500$ $m = 100$	$h = 50$	$h = 100$	$h = 200$
Theta Forecaster	0.54	0.46	0.44
Linear Regression	0.63	0.55	0.46
ARIMA	5.15	4.37	4.14
DeepAR	35.72	54.93	134.13
Random Forest	2.39	3.58	4.38
LightGBM	0.57	1.10	0.98
Logistic Regression	2.41	2.99	2.01
Ordinal Regression	1.36	1.30	1.43
Pairwise Ranking	1.90	2.20	2.65
DynaPairwise Ranking	3.17	3.31	3.40

are not significantly positive in a market cycle, which signals a strong mean-reversion style of the asset class.

5.4.3 Experimental setting

All results of both competing and proposed methods are produced based on a common benchmark procedure. For each of the retrieved time series, we retrained on a rolling basis, i.e., every N_{roll} trading days and varied the forecasting horizon from 50 trading days to 200 trading days accordingly. By default, the forward-looking forecast horizon h is aligned with the interval of the rolling retraining by setting $h = N_{\text{roll}}$. By varying the horizon from short term $h = 50$ to mid-long term $h = \{100, 200\}$, we validate the consistency of the results and report the evaluation metrics together with their statistical significance, i.e. standard deviation. To summarize, the benchmark procedure is analogous to Algorithm 1, where the LL in Algorithm 2 and the iterative dynamic prediction in Algorithm 3 are not considered.

Table 5.4: A summary of predictive performance on equity indices and fixed income indices, measured by tracking errors and relative relationships, i.e., Symmetric mean absolute percentage error (SMAPE), Root-mean-square error (RMSE) and Concordance index (CI). ©IEEE, 2021

$N_{\textbf{train}} = 500$		$h = 50$			$h = 100$			$h = 200$		
$m = 100$		**SMAPE**	**RMSE**	**CI**	**SMAPE**	**RMSE**	**CI**	**SMAPE**	**RMSE**	**CI**
S&P 500	Theta Forecaster	0.02 (0.01)	55.86	0.55 (0.06)	0.03 (0.01)	81.21	0.51 (0.04)	0.04 (0.01)	110.72	0.51 (0.01)
	Linear Regression	0.02 (0.01)	53.01	0.64 (0.07)	0.03 (0.01)	84.53	0.60 (0.06)	0.05 (0.02)	129.65	0.61 (0.07)
	ARIMA	0.02 (0.01)	**51.12**	0.51 (0.02)	0.03 (0.01)	**73.58**	0.51 (0.01)	0.04 (0.02)	**107.55**	0.52 (0.03)
	DeepAR	0.03 (0.02)	58.10	0.52 (0.05)	0.06 (0.04)	86.55	0.52 (0.05)	0.06 (0.03)	143.26	0.53 (0.03)
	Random Forest	0.02 (0.01)	56.83	0.68 (0.14)	0.04 (0.01)	84.94	**0.72 (0.11)**	0.06 (0.02)	141.48	0.76 (0.14)
	LightGBM	-	-	0.62 (0.20)	-	-	0.72 (0.24)	-	-	0.77 (0.23)
	Logistic Regression	-	-	0.59 (0.17)	-	-	0.60 (0.18)	-	-	**0.80 (0.20)**
	Ordinal Regression	-	-	0.62 (0.20)	-	-	0.62 (0.20)	-	-	0.65 (0.22)
	Pairwise Ranking	-	-	0.63 (0.08)	-	-	0.61 (0.06)	-	-	0.62 (0.07)
	DynaPairwise Ranking	0.03 (0.02)	75.94	**0.73 (0.13)**	0.04 (0.01)	94.53	0.70 (0.15)	0.06 (0.02)	150.89	0.73 (0.14)
Russell 2000	Theta Forecaster	0.03 (0.01)	43.95	0.56 (0.06)	0.04 (0.02)	61.85	0.53 (0.05)	0.06 (0.02)	87.82	0.52 (0.04)
	Linear Regression	0.02 (0.01)	40.98	0.63 (0.09)	0.04 (0.02)	58.18	0.62 (0.05)	0.07 (0.03)	95.96	0.60 (0.11)
	ARIMA	0.03 (0.01)	**37.79**	0.53 (0.06)	0.04 (0.02)	**51.56**	0.54 (0.11)	0.06 (0.02)	**82.43**	0.55 (0.12)
	DeepAR	0.02 (0.02)	49.46	0.52 (0.06)	0.05 (0.03)	68.84	0.52 (0.04)	0.08 (0.4)	104.30	0.52 (0.01)
	Random Forest	0.03 (0.02)	46.26	0.66 (0.09)	0.05 (0.02)	62.80	0.68 (0.11)	0.07 (0.02)	92.78	0.67 (0.16)
	LightGBM	-	-	0.65 (0.21)	-	-	0.63 (0.20)	-	-	0.64 (0.22)
	Logistic Regression	-	-	0.61 (0.17)	-	-	0.65 (0.19)	-	-	**0.75 (0.22)**
	Ordinal Regression	-	-	0.59 (0.17)	-	-	0.63 (0.20)	-	-	0.70 (0.20)
	Pairwise Ranking	-	-	0.63 (0.08)	-	-	0.60 (0.08)	-	-	0.61 (0.10)
	DynaPairwise Ranking	0.04 (0.02)	57.99	**0.70 (0.11)**	0.05 (0.02)	73.30	**0.70 (0.11)**	0.09 (0.04)	124.69	0.65 (0.09)
DAX	Theta Forecaster	0.03 (0.02)	448.49	0.57 (0.06)	0.05 (0.02)	652.03	0.54 (0.04)	0.07 (0.03)	887.01	0.54 (0.02)
	Linear Regression	0.03 (0.01)	**410.94**	0.62 (0.08)	0.04 (0.02)	611.71	0.57 (0.05)	0.07 (0.03)	868.62	0.58 (0.07)
	ARIMA	0.03 (0.02)	419.74	0.54 (0.10)	0.05 (0.03)	**600.70**	0.53 (0.09)	0.07 (0.03)	**815.33**	0.57 (0.12)
	DeepAR	0.05 (0.02)	503.56	0.51 (0.06)	0.07 (0.04)	798.61	0.52 (0.04)	0.08 (0.04)	908.40	0.51 (0.03)
	Random Forest	0.04 (0.02)	427.67	0.68 (0.12)	0.06 (0.04)	650.77	0.68 (0.11)	0.08 (0.04)	882.51	0.71 (0.12)
	LightGBM	-	-	0.62 (0.19)	-	-	0.63 (0.21)	-	-	0.67 (0.19)
	Logistic Regression	-	-	0.60 (0.17)	-	-	0.63 (0.18)	-	-	**0.76 (0.17)**
	Ordinal Regression	-	-	0.54 (0.12)	-	-	0.57 (0.17)	-	-	0.57 (0.17)
	Pairwise Ranking	-	-	0.61 (0.07)	-	-	0.61 (0.06)	-	-	0.60 (0.08)
	DynaPairwise Ranking	0.05 (0.03)	721.36	**0.75 (0.10)**	0.06 (0.03)	830.78	**0.68 (0.14)**	0.09 (0.04)	1106.25	0.66 (0.16)
TOPIX	Theta Forecaster	0.04 (0.02)	71.39	0.60 (0.09)	0.06 (0.03)	111.43	0.58 (0.05)	0.12 (0.07)	192.67	0.58 (0.05)
	Linear Regression	0.03 (0.02)	67.03	0.64 (0.12)	0.06 (0.04)	108.99	0.62 (0.10)	0.12 (0.07)	189.64	0.63 (0.07)
	ARIMA	0.04 (0.03)	**63.98**	0.64 (0.17)	0.06 (0.03)	**102.09**	0.59 (0.13)	0.12 (0.06)	176.59	0.63 (0.14)
	DeepAR	0.05 (0.02)	83.52	0.53 (0.06)	0.08 (0.05)	133.58	0.51 (0.02)	0.09 (0.04)	**154.85**	0.51 (0.03)
	Random Forest	0.04 (0.02)	66.43	0.69 (0.11)	0.06 (0.04)	103.50	**0.74 (0.14)**	0.12 (0.06)	182.72	0.68 (0.17)
	LightGBM	-	-	0.66 (0.21)	-	-	0.61 (0.20)	-	-	**0.76 (0.23)**
	Logistic Regression	-	-	0.61 (0.17)	-	-	0.64 (0.21)	-	-	0.74 (0.17)
	Ordinal Regression	-	-	0.55 (0.15)	-	-	0.60 (0.18)	-	-	0.70 (0.23)
	Pairwise Ranking	-	-	0.66 (0.10)	-	-	0.63 (0.08)	-	-	0.64 (0.11)
	DynaPairwise Ranking	0.05 (0.03)	99.35	**0.70 (0.12)**	0.07 (0.03)	122.89	0.66 (0.10)	0.14 (0.05)	216.19	0.68 (0.11)
Bund 10Y	Theta Forecaster	0.01 (0.01)	2.96	0.56 (0.07)	0.01 (0.01)	4.19	0.53 (0.05)	0.02 (0.01)	**6.07**	0.51 (0.03)
	Linear Regression	0.01 (0.01)	2.97	0.64 (0.08)	0.02 (0.01)	4.48	0.62 (0.08)	0.02 (0.01)	6.55	0.61 (0.10)
	ARIMA	0.01 (0.01)	2.48	0.53 (0.06)	0.02 (0.01)	**3.73**	0.53 (0.08)	0.03 (0.02)	6.47	0.54 (0.08)
	DeepAR	0.01 (0.01)	**2.45**	0.51 (0.06)	0.02 (0.02)	4.35	0.51 (0.04)	0.04 (0.03)	9.24	0.51 (0.04)
	Random Forest	0.01 (0.01)	2.74	0.72 (0.13)	0.02 (0.01)	4.27	0.69 (0.15)	0.03 (0.02)	6.13	0.67 (0.18)
	LightGBM	-	-	0.61 (0.21)	-	-	0.61 (0.21)	-	-	0.57 (0.17)
	Logistic Regression	-	-	0.56 (0.15)	-	-	0.69 (0.21)	-	-	0.68 (0.21)
	Ordinal Regression	-	-	0.54 (0.12)	-	-	0.57 (0.16)	-	-	0.56 (0.15)
	Pairwise Ranking	-	-	0.64 (0.09)	-	-	0.56 (0.06)	-	-	0.56 (0.09)
	DynaPairwise Ranking	0.01 (0.01)	3.82	**0.72 (0.11)**	0.02 (0.01)	5.28	**0.69 (0.14)**	0.03 (0.01)	7.46	**0.70 (0.18)**
Gilt 10Y	Theta Forecaster	0.01 (0.01)	3.82	0.56 (0.07)	0.02 (0.01)	5.28	0.55 (0.08)	0.03 (0.02)	7.46	0.54 (0.09)
	Linear Regression	0.01 (0.01)	3.20	0.64 (0.05)	0.02 (0.01)	3.92	0.63 (0.07)	0.03 (0.02)	7.42	0.63 (0.11)
	ARIMA	0.01 (0.01)	3.10	0.55 (0.10)	0.02 (0.01)	**3.46**	0.52 (0.05)	0.02 (0.02)	**5.43**	0.52 (0.06)
	DeepAR	0.01 (0.01)	**2.97**	0.52 (0.05)	0.03 (0.02)	5.57	0.50 (0.04)	0.05 (0.03)	9.08	0.52 (0.02)
	Random Forest	0.02 (0.01)	3.30	**0.70 (0.11)**	0.02 (0.01)	4.26	0.67 (0.11)	0.03 (0.02)	6.19	0.65 (0.09)
	LightGBM	-	-	0.65 (0.22)	-	-	**0.69 (0.24)**	-	-	0.57 (0.17)
	Logistic Regression	-	-	0.58 (0.16)	-	-	0.64 (0.21)	-	-	**0.70 (0.23)**
	Ordinal Regression	-	-	0.60 (0.16)	-	-	0.60 (0.18)	-	-	0.63 (0.21)
	Pairwise Ranking	-	-	0.63 (0.08)	-	-	0.64 (0.11)	-	-	0.63 (0.13)
	DynaPairwise Ranking	0.02 (0.01)	4.45	0.68 (0.08)	0.02 (0.01)	5.49	0.66 (0.10)	0.04 (0.03)	9.26	0.60 (0.07)
Treasury 10Y	Theta Forecaster	0.01 (0.01)	2.38	0.61 (0.10)	0.01 (0.01)	3.22	0.60 (0.10)	0.02 (0.01)	4.37	0.58 (0.10)
	Linear Regression	0.01 (0.00)	**2.14**	0.65 (0.09)	0.01 (0.00)	**2.80**	0.63 (0.07)	0.01 (0.01)	3.79	0.60 (0.10)
	ARIMA	0.01 (0.00)	2.19	0.67 (0.16)	0.01 (0.01)	2.90	0.66 (0.18)	0.02 (0.01)	3.75	0.63 (0.17)
	DeepAR	0.01 (0.01)	2.22	0.52 (0.04)	0.02 (0.01)	4.54	0.52 (0.03)	0.04 (0.03)	10.02	0.51 (0.03)
	Random Forest	0.01 (0.01)	2.44	0.67 (0.11)	0.02 (0.01)	3.43	0.65 (0.09)	0.02 (0.01)	**3.70**	**0.70 (0.09)**
	LightGBM	-	-	0.64 (0.20)	-	-	0.64 (0.19)	-	-	0.50 (0.00)
	Logistic Regression	-	-	0.57 (0.15)	-	-	0.56 (0.11)	-	-	0.65 (0.21)
	Ordinal Regression	-	-	0.56 (0.14)	-	-	0.56 (0.15)	-	-	0.59 (0.16)
	Pairwise Ranking	-	-	0.65 (0.10)	-	-	0.63 (0.06)	-	-	0.58 (0.08)
	DynaPairwise Ranking	0.01 (0.01)	3.22	**0.70 (0.11)**	0.02 (0.01)	4.06	**0.66 (0.11)**	0.02 (0.01)	5.27	0.65 (0.11)

Table 5.5: A summary of predictive performance on currency time series, measured by tracking errors and relative relationships, i.e., Symmetric mean absolute percentage error (SMAPE), Root-mean-square error (RMSE) and Concordance index (CI). ©IEEE, 2021

$N_{\textbf{train}} = 500$		$h = 50$			$h = 100$			$h = 200$		
$m = 100$		**SMAPE**	**RMSE**	**CI**	**SMAPE**	**RMSE**	**CI**	**SMAPE**	**RMSE**	**CI**
EURUSD	Theta Forecaster	0.03 (0.02)	4.63	0.61 (0.08)	0.04 (0.03)	6.70	0.60 (0.10)	0.05 (0.02)	7.14	0.62 (0.10)
	Linear Regression	0.03 (0.02)	4.34	0.64 (0.08)	0.04 (0.03)	6.24	0.60 (0.06)	0.05 (0.02)	6.84	0.60 (0.06)
	ARIMA	0.03 (0.03)	3.99	0.56 (0.15)	0.05 (0.04)	6.13	0.57 (0.15)	0.07 (0.08)	9.10	0.62 (0.20)
	DeepAR	0.04 (0.03)	4.88	0.52 (0.05)	0.06 (0.04)	7.79	0.49 (0.04)	0.09 (0.08)	11.22	0.51 (0.03)
	Random Forest	0.03 (0.02)	**3.85**	0.66 (0.09)	0.04 (0.03)	**5.52**	0.64 (0.08)	0.04 (0.02)	**5.16**	0.64 (0.10)
	LightGBM	-	-	0.70 (0.22)	-	-	0.68 (0.23)	-	-	0.69 (0.22)
	Logistic Regression	-	-	0.59 (0.16)	-	-	0.69 (0.20)	-	-	0.75 (0.18)
	Ordinal Regression	-	-	0.62 (0.18)	-	-	0.67 (0.19)	-	-	0.53 (0.08)
	Pairwise Ranking	-	-	0.65 (0.08)	-	-	0.63 (0.09)	-	-	0.64 (0.07)
	DynaPairwise Ranking	0.03 (0.02)	4.86	**0.78 (0.14)**	0.04 (0.03)	6.66	**0.77 (0.18)**	0.05 (0.02)	7.35	**0.77 (0.12)**
GBPUSD	Theta Forecaster	0.03 (0.02)	4.21	0.65 (0.11)	0.04 (0.04)	6.78	0.62 (0.11)	0.04 (0.02)	7.10	0.63 (0.09)
	Linear Regression	0.02 (0.02)	4.02	0.67 (0.07)	0.04 (0.03)	6.25	0.63 (0.08)	0.04 (0.02)	6.65	0.60 (0.08)
	ARIMA	0.03 (0.02)	3.33	0.54 (0.10)	0.04 (0.04)	5.54	0.57 (0.13)	0.08 (0.10)	9.81	0.60 (0.16)
	DeepAR	0.03 (0.03)	3.57	0.51 (0.06)	0.05 (0.04)	7.00	0.52 (0.05)	0.10 (0.09)	11.68	0.51 (0.04)
	Random Forest	0.02 (0.02)	**3.13**	0.66 (0.09)	0.04 (0.03)	**5.29**	0.63 (0.10)	0.04 (0.02)	**5.33**	0.65 (0.10)
	LightGBM	-	-	0.71 (0.22)	-	-	0.73 (0.22)	-	-	0.83 (0.20)
	Logistic Regression	-	-	0.62 (0.19)	-	-	0.64 (0.20)	-	-	0.71 (0.20)
	Ordinal Regression	-	-	0.55 (0.13)	-	-	0.58 (0.17)	-	-	0.55 (0.14)
	Pairwise Ranking	-	-	0.65 (0.08)	-	-	0.64 (0.09)	-	-	0.63 (0.09)
	DynaPairwise Ranking	0.03 (0.02)	4.29	**0.76 (0.13)**	0.04 (0.03)	6.69	**0.79 (0.14)**	0.04 (0.02)	6.72	**0.86 (0.09)**
CHFUSD	Theta Forecaster	0.03 (0.02)	5.41	0.58 (0.07)	0.05 (0.04)	8.39	0.54 (0.38)	0.06 (0.03)	8.12	0.56 (0.07)
	Linear Regression	0.03 (0.03)	5.61	0.63 (0.09)	0.05 (0.04)	8.76	0.59 (0.09)	0.05 (0.03)	7.64	0.59 (0.07)
	ARIMA	0.03 (0.03)	4.33	0.56 (0.10)	0.06 (0.06)	8.02	0.53 (0.05)	0.06 (0.04)	7.86	0.54 (0.06)
	DeepAR	0.03 (0.03)	5.28	0.52 (0.07)	0.07 (0.04)	9.52	0.51 (0.04)	0.14 (0.12)	15.06	0.50 (0.02)
	Random Forest	0.03 (0.02)	**4.11**	0.69 (0.11)	0.05 (0.03)	**6.73**	0.65 (0.10)	0.06 (0.03)	**7.51**	0.65 (0.14)
	LightGBM	-	-	0.70 (0.21)	-	-	0.73 (0.20)	-	-	0.75 (0.20)
	Logistic Regression	-	-	0.62 (0.18)	-	-	0.67 (0.21)	-	-	0.73 (0.21)
	Ordinal Regression	-	-	0.66 (0.20)	-	-	0.68 (0.22)	-	-	0.75 (0.21)
	Pairwise Ranking	-	-	0.64 (0.07)	-	-	0.61 (0.08)	-	-	0.62 (0.09)
	DynaPairwise Ranking	0.03 (0.02)	5.63	**0.73 (0.13)**	0.05 (0.04)	8.36	**0.73 (0.18)**	0.05 (0.03)	8.13	**0.76 (0.19)**
JPYUSD	Theta Forecaster	0.02 (0.02)	2.77	0.63 (0.11)	0.03 (0.02)	4.06	0.59 (0.10)	0.05 (0.04)	6.73	0.57 (0.09)
	Linear Regression	0.02 (0.02)	2.76	0.62 (0.07)	0.03 (0.02)	4.03	0.57 (0.06)	0.05 (0.04)	6.78	0.56 (0.05)
	ARIMA	0.02 (0.02)	**2.45**	0.53 (0.07)	0.04 (0.02)	3.67	0.51 (0.02)	0.05 (0.04)	5.54	0.51 (0.03)
	DeepAR	0.03 (0.02)	3.12	0.54 (0.04)	0.04 (0.03)	3.90	0.49 (0.03)	0.14 (0.08)	13.75	0.51 (0.03)
	Random Forest	0.02 (0.02)	2.55	0.69 (0.11)	0.03 (0.02)	**3.59**	0.69 (0.13)	0.05 (0.04)	**5.27**	0.68 (0.13)
	LightGBM	-	-	0.67 (0.22)	-	-	0.65 (0.22)	-	-	0.68 (0.23)
	Logistic Regression	-	-	0.60 (0.18)	-	-	0.60 (0.17)	-	-	0.59 (0.14)
	Ordinal Regression	-	-	0.56 (0.13)	-	-	0.62 (0.19)	-	-	0.58 (0.16)
	Pairwise Ranking	-	-	0.62 (0.09)	-	-	0.60 (0.08)	-	-	0.60 (0.08)
	DynaPairwise Ranking	0.03 (0.02)	3.64	**0.73 (0.14)**	0.04 (0.02)	4.84	**0.83 (0.12)**	0.06 (0.04)	7.07	**0.83 (0.08)**
CADUSD	Theta Forecaster	0.02 (0.02)	5.13	0.63 (0.10)	0.03 (0.03)	6.73	0.60 (0.09)	0.03 (0.02)	7.38	0.57 (0.09)
	Linear Regression	0.02 (0.02)	4.92	0.63 (0.08)	0.03 (0.02)	6.19	0.61 (0.10)	0.03 (0.02)	6.44	0.57 (0.10)
	ARIMA	0.02 (0.02)	**3.67**	0.55 (0.11)	0.03 (0.03)	5.22	0.55 (0.13)	0.04 (0.02)	6.98	0.65 (0.21)
	DeepAR	0.02 (0.03)	4.04	0.52 (0.05)	0.04 (0.02)	6.60	0.51 (0.04)	0.09 (0.05)	16.16	0.50 (0.02)
	Random Forest	0.02 (0.02)	3.78	0.65 (0.09)	0.03 (0.02)	**5.22**	0.67 (0.12)	0.03 (0.02)	**6.38**	0.69 (0.11)
	LightGBM	-	-	0.65 (0.20)	-	-	0.62 (0.19)	-	-	0.68 (0.23)
	Logistic Regression	-	-	0.63 (0.19)	-	-	0.69 (0.21)	-	-	0.76 (0.20)
	Ordinal Regression	-	-	0.62 (0.18)	-	-	0.71 (0.22)	-	-	**0.87 (0.20)**
	Pairwise Ranking	-	-	0.64 (0.09)	-	-	0.61 (0.10)	-	-	0.57 (0.07)
	DynaPairwise Ranking	0.03 (0.02)	5.19	**0.78 (0.12)**	0.03 (0.02)	6.77	**0.78 (0.13)**	0.04 (0.05)	10.01	0.85 (0.11)
AUDUSD	Theta Forecaster	0.04 (0.04)	9.75	0.60 (0.07)	0.05 (0.05)	14.76	0.58 (0.06)	0.06 (0.04)	15.61	0.54 (0.05)
	Linear Regression	0.03 (0.03)	9.43	0.63 (0.10)	0.05 (0.05)	14.25	0.59 (0.08)	0.06 (0.04)	15.88	0.59 (0.08)
	ARIMA	0.04 (0.04)	**8.09**	0.60 (0.15)	0.07 (0.07)	13.40	0.59 (0.15)	0.11 (0.14)	21.88	0.56 (0.16)
	DeepAR	0.04 (0.04)	8.79	0.53 (0.06)	0.09 (0.06)	18.57	0.53 (0.04)	0.12 (0.05)	26.64	0.53 (0.02)
	Random Forest	0.04 (0.04)	8.17	0.70 (0.08)	0.06 (0.06)	**12.41**	0.68 (0.11)	0.07 (0.03)	**14.72**	0.70 (0.11)
	LightGBM	-	-	0.62 (0.19)	-	-	0.70 (0.21)	-	-	**0.77 (0.18)**
	Logistic Regression	-	-	0.57 (0.15)	-	-	0.55 (0.11)	-	-	0.55 (0.13)
	Ordinal Regression	-	-	0.58 (0.16)	-	-	0.63 (0.18)	-	-	0.69 (0.22)
	Pairwise Ranking	-	-	0.66 (0.11)	-	-	0.61 (0.10)	-	-	0.63 (0.09)
	DynaPairwise Ranking	0.04 (0.03)	11.42	**0.79 (0.13)**	0.06 (0.05)	15.72	**0.75 (0.14)**	0.07 (0.04)	17.97	0.71 (0.17)
NZDUSD	Theta Forecaster	0.04 (0.03)	9.32	0.58 (0.07)	0.05 (0.04)	14.17	0.55 (0.06)	0.07 (0.03)	17.03	0.53 (0.05)
	Linear Regression	0.03 (0.03)	8.44	0.64 (0.08)	0.05 (0.04)	13.16	0.59 (0.07)	0.07 (0.04)	16.80	0.62 (0.08)
	ARIMA	0.04 (0.03)	**8.16**	0.55 (0.11)	0.07 (0.05)	13.61	0.55 (0.12)	0.12 (0.13)	22.62	0.54 (0.11)
	DeepAR	0.04 (0.03)	9.10	0.52 (0.06)	0.06 (0.05)	13.50	0.51 (0.04)	0.20 (0.09)	39.06	0.50 (0.01)
	Random Forest	0.04 (0.04)	8.52	0.69 (0.12)	0.06 (0.05)	**12.49**	0.68 (0.14)	0.07 (0.02)	**15.64**	0.66 (0.09)
	LightGBM	-	-	0.59 (0.17)	-	-	0.68 (0.22)	-	-	0.76 (0.21)
	Logistic Regression	-	-	0.62 (0.19)	-	-	0.65 (0.19)	-	-	**0.77 (0.20)**
	Ordinal Regression	-	-	0.56 (0.14)	-	-	0.55 (0.13)	-	-	0.55 (0.14)
	Pairwise Ranking	-	-	0.66 (0.08)	-	-	0.63 (0.07)	-	-	0.60 (0.09)
	DynaPairwise Ranking	0.04 (0.03)	10.97	**0.75 (0.14)**	0.05 (0.04)	14.11	**0.78 (0.13)**	0.07 (0.03)	16.73	0.68 (0.17)
NOKUSD	Theta Forecaster	0.03 (0.03)	7.08	0.60 (0.09)	0.05 (0.04)	10.41	0.57 (0.08)	0.05 (0.02)	10.13	0.54 (0.05)
	Linear Regression	0.03 (0.03)	6.62	0.64 (0.09)	0.04 (0.03)	9.68	0.59 (0.07)	0.05 (0.02)	9.76	0.60 (0.08)
	ARIMA	0.03 (0.03)	5.83	0.60 (0.17)	0.05 (0.05)	9.12	0.62 (0.19)	0.08 (0.12)	12.63	0.74 (0.20)
	DeepAR	0.03 (0.03)	5.76	0.53 (0.06)	0.06 (0.04)	9.63	0.50 (0.03)	0.13 (0.11)	18.61	0.49 (0.02)
	Random Forest	0.03 (0.03)	**5.57**	0.66 (0.12)	0.05 (0.03)	**8.00**	0.64 (0.10)	0.05 (0.02)	**8.54**	0.63 (0.13)
	LightGBM	-	-	0.63 (0.20)	-	-	0.63 (0.19)	-	-	0.73 (0.23)
	Logistic Regression	-	-	0.65 (0.18)	-	-	0.74 (0.20)	-	-	0.77 (0.20)
	Ordinal Regression	-	-	0.62 (0.19)	-	-	0.61 (0.20)	-	-	0.59 (0.15)
	Pairwise Ranking	-	-	0.62 (0.09)	-	-	0.59 (0.07)	-	-	0.59 (0.06)
	DynaPairwise Ranking	0.04 (0.02)	7.53	**0.75 (0.13)**	0.06 (0.04)	11.93	**0.80 (0.14)**	0.07 (0.06)	15.24	**0.84 (0.10)**
SEKUSD	Theta Forecaster	0.04 (0.03)	5.56	0.59 (0.08)	0.05 (0.04)	7.95	0.57 (0.07)	0.06 (0.03)	8.82	0.56 (0.06)
	Linear Regression	0.03 (0.03)	5.15	0.62 (0.07)	0.04 (0.04)	7.37	0.58 (0.08)	0.05 (0.03)	8.11	0.58 (0.07)
	ARIMA	0.04 (0.03)	**4.71**	0.53 (0.10)	0.06 (0.05)	7.04	0.53 (0.09)	0.10 (0.13)	12.00	0.58 (0.15)
	DeepAR	0.04 (0.04)	5.41	0.51 (0.04)	0.08 (0.05)	10.25	0.51 (0.04)	0.09 (0.05)	11.50	0.48 (0.02)
	Random Forest	0.04 (0.03)	4.81	0.67 (0.10)	0.05 (0.04)	**6.22**	0.69 (0.11)	0.06 (0.04)	**7.51**	0.65 (0.12)
	LightGBM	-	-	0.64 (0.20)	-	-	0.67 (0.21)	-	-	0.70 (0.21)
	Logistic Regression	-	-	0.62 (0.19)	-	-	0.70 (0.20)	-	-	0.74 (0.22)
	Ordinal Regression	-	-	0.64 (0.20)	-	-	0.69 (0.21)	-	-	0.67 (0.20)
	Pairwise Ranking	-	-	0.63 (0.08)	-	-	0.59 (0.08)	-	-	0.59 (0.07)
	DynaPairwise Ranking	0.04 (0.03)	5.64	**0.75 (0.13)**	0.05 (0.04)	8.31	**0.80 (0.14)**	0.05 (0.03)	8.56	**0.82 (0.11)**

For each of the individual time series, we take the last 30% of the retrieved data as the test set and report the performance based on the forecasted values in that horizon. Given the benchmark scheme that behaves in a rolling manner, we set the latest $N_{\text{train}} = 500$ trading days as the input data for all the models involved and set $m = N_{\text{train}}$ for Algorithm 2 by default, so that all observed data are exploited in LL. In practice, the results delivered by setting $m = 100$, $m = 500$ and $m = 1,000$ does not show a significant difference; therefore, we report all results by setting $m = N_{\text{train}}$ as the default configuration. The detailed hyperparameters for the benchmark procedure and the proposed learning algorithm are summarized in Table 5.1.

In practice, we follow the standard de facto hyperparameter tuning processes and models available in packages like scikit-learn library [131], sktime [174] and GluonTS [175]. The CPU time of each method taken for the cross-validation is based on the computation on a Macbook Pro with 2.9 GHz Dual-Core Intel Core i5 and 16 GB 1867 MHz DDR3 memory. A summary of the CPU time for model training is summarized in Table 5.3. Note that although the averaged CPU time for training a model in cross-validation is obtained by using different implementations in scikit-learn, sktime and GluonTS, a comparison between the CPU time for each method is meaningful, because for each length of the forward-looking horizon, the number of models to be trained is fixed. Given a fixed h, the most time-efficient method is Theta Forecaster for all lengths of forward-looking horizons. The proposed method is moderate among others. The highest CPU time is reported from the experiments on the DeepAR method, for which the training is subject to a termination after 5 epochs which is set empirically.

5.5 Discussions

In this section, we discuss the experimental results on real world financial time series. In the first part, we examine the use of the proposed methods from an alternative perspective other than tracking error; specifically, we examine the impact of the proposed method in the lens of time to event (TTE), which is critical in certain financial applications [164, 167]. In the second part, we discuss the practical issues, such as the performance of the proposed methods on different types of time series and under different lengths of the forecast horizon. The details of the results are presented in Table 5.4 and Table 5.5.

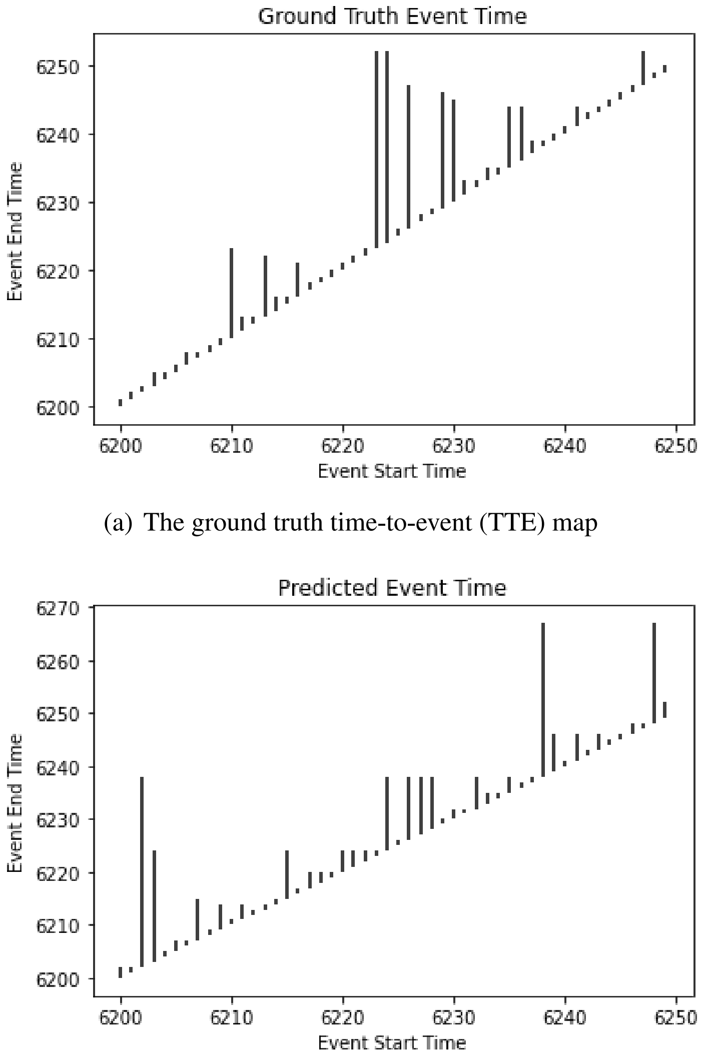
5.5.1 Impacts of concordance index

The CI is a well-established evaluation metric for survival analysis in medical statistics [144]. The metric addresses the differences along the timeline between a forecasted event and a ground truth event. It essentially indicates the time until the occurrence of an event of interest (EOI), e.g., death, onset of a disease, or failure of a machine, depending on the domain. The adoption of CI as the major evaluation metrics instead of the tracking errors, happens in a wide spectrum of applications from clinical research [141], epidemiology, disease control [170] to predictive maintenance [107], reliability engineering [171], and insurance [172]. We hereby discuss its property, usage and potential impacts in financial applications.

Recalling the definition of the CI in Equation (5.19), TTE in Figure 5.5, and the gap between its ground truth and its forecast by the ranking model can be written as

$$\begin{aligned}
\text{TTE}(i) &= \sum_{\substack{\forall r_j > r_i \\ \forall j > i}} \min(j-i) && (5.23)\\
\widehat{\text{TTE}}(i) &= \sum_{\substack{\forall \hat{r}_j > \hat{r}_i \\ \forall j > i}} \min(j-i) && (5.24)\\
\Delta\text{TTE}(i) &= \left|\text{TTE}(i) - \widehat{\text{TTE}}(i)\right| && (5.25)
\end{aligned}$$

where TTE(i) and $\widehat{\text{TTE}}(i)$ refer to the length of the vertical black lines in Figure 5.5. A permutation of the neighboring items in the ground truth ranking list would lead to a same directional change of ΔTTE(i) and CI. When the CI is maximized to 1, ΔTTE(i) is minimized to 0. A higher CI score indicates not only the correctness of the relative ordering of the multi-step ahead forecast but also the similarity between the two TTE maps in Figure 5.5.



(a) The ground truth time-to-event (TTE) map

(b) The forecasted time-to-event (TTE) map by the proposed method

Figure 5.5: The time-to-event (TTE) alignment between the ground truth and the forecast by the proposed method. The event of interest (EOI) is defined as the successful observation of a higher scalar. The horizontal index indicates the starting time epoch of each event, and the vertical index indicates the time epoch when an EOI is first observed. Therefore, the length of the black bars in the figure indicates the duration before an EOI occurs. The horizon of multi-step ahead forecasting is set to 50, and by forecasting 100 time epochs, we obtain a TTE estimate without censoring. ©IEEE, 2021

In the financial domain, the similarity between the ground truth TTE map and the forecasted TTE map is important. The existence of certain financial derivative instruments enable the monetization of the forecasted TTE map by the proposed models. Typical examples include future contracts, which can be used to construct a long-short investment portfolio [94, 88, 87]. With proper position sizing [90] and risk management, the signals or forecast derived from the model can be transformed into an investment strategy. Following the intuition that a future contract has a profit and loss profile that is dependent on the price difference at two different time epochs, we define the EOI as the successful observation of a higher value in the time series within a specified forward-looking horizon h from time i. As depicted in Figure 5.5, with more accurate forecasting of the TTE map, better investment decisions for hunting alternative returns from the financial asset class can be made easily. By refering to a TTE map, one can answer questions such as, should the current position be renewed at its termination and by how much.

The relationship between the CI and the TTE is intuitive. In Figure 5.5, the ground truth TTE map and the forecasted TTE map are visualized. The horizontal index indicates the starting time epoch of each event, and the vertical index indicates the time epoch when an EOI is first observed. The length of the black bars in the figure indicates the duration before an EOI occurs. The horizon of multi-step ahead forecasting is set to 50, and by forecasting 100 time epochs, we obtain a time-to-event estimate without censoring. In essence, the CI is a generalization of the gap between the ground truth TTE map in Figure 5.5-a) and the forecasted TTE map in Figure 5.5-b). Given a cross-temporal pair (i, j), TTE neglects the impact of the forecasted order during $t \in [i + 1 : j - 1]$, if they are correct with respect to $\hat{r}_t$, the forecasted order at time j.

Overall, although it is technically difficult to directly optimize such a gap denoted by ΔTTE(i), we argue that the proposed learning-to-rank approach which intends to optimize the concordance at a pairwise level is a promising solution for time-series forecasting, especially when relative ordering within a specified horizon matters to the domain experts.

5.5.2 Growth index vs. Mean-reverting index

As shown in Table 5.4 and Table 5.5, the performance of the proposed method varies by the underlying asset class of the indices. For the growth indices in Table 5.4, we vary the forward-looking horizon of the competing models. In terms of our optimisation objective, i.e. CI, the proposed dynamic prediction scheme outperforms its competing methods in the short-term settings, where $h = 50$ and 100, and underperforms its competing methods in the long-term settings, where $h = 200$. Among the competing methods, the ARIMA model, the DeepAR model and the random forest model are the ones that deliver superior performance in terms of tracking error and deserve an attention. The DeepAR model outperformed consistently in fixed income indices, e.g. German 10Y Bund and UK 10Y Gilt. Given the low volatility of these fixed income indices recorded in Table 5.2, we think that the DeepAR method outperforms because the method captures the short-term trends better than others by leveraging the recurrent neural networks. However, the other side of this superior tracking error performance is its poor performance in terms of relative ordering or CI. Recalling that the growth indices share a common positive expected upward movement, we adopt Equation (5.22) to calibrate this expectation in reference to Table 5.2. However, such a calibration is trivial in short-term and does not show a significant impact in long-term forecasting with $h = 200$. We argue that this is partially due to the fact that the calibration is not integrated into the learning phase. When the length of the forward-looking horizon increases, the discriminative power of our model decreases dramatically. Similar deterioration is also observed in the results by the competing methods.

In contrast to this significant deterioration, the performance deterioration with respect to the increase in the length of the forward-looking horizon is not significant for mean-reverting indices except for the ARIMA model and the DeepAR model, as observed in Table 5.5. We attribute this advantage of the proposed method to the increased volatility of the mean-reverting time series. The mean-reverting indices have on average higher standard deviations in terms of rolling return; therefore given a fixed length of horizon, both the features and the observations in the time series are more informative and discriminative. Since the momentum factor does not function well for mean-reverting time series, we also observed that among the competing methods, the random forest model performs consistently better than the ARIMA model and the DeepAR model. This is counterfactual to what we observed in Table 5.4, where the linear regression, the random forest model, the ARIMA model and the DeepAR model tie in terms of tracking error metrics, e.g., SMAPE and RMSE. We argue that, this is because the moving average module in ARIMA detracts the performance by explicitly incorporating the momentum, which is meaningless for mean-reverting time series.

Overall, the proposed model outperforms its competing methods for all mean-reverting indices in almost all settings of the forward-looking horizon. This implies that the proposed dynamic prediction scheme is a better solution in situations where the utility of the forecast is dependent on the relative ordering rather than on tracking error; the forecasting horizon is long; and the time series exhibits high volatility or strong mean-reverting behavior.

Note that although the focus of this work is the relative ordering of multi-step ahead forecasting. The proposed dynamic prediction scheme has also achieved comparable performance in terms of tracking error, i.e., SMAPE and RMSE, while significantly excelling its competing methods in terms of the CI. For a time series in Table 5.4 and Table 5.5, an improvement in the tracking error is not empirically correlated with the improvement in the relative ordering metrics. We argue that, the tracking error metrics and the relative ordering metrics which is advocated in the proposed method are orthogonal or even complementary to each other. In the DeepAR method, a best RMSE score is likely associated with a worst CI score, like for German Bund 10Y and for UK Gilt 10Y. With the promising experimental results by the proposed method in terms of CI score, we argue that the differences and connections between the competing methods and the proposed method worth a thorough investigation in the future.

5.6 Summary

Data veracity of time series refers to the uncertainty of sequential observations. Relative ordering of the noisy sequential observations is a robust interpretation, which can be utilized to develop promising algorithms that directly tackle the data veracity of time series. We proposed a learning-to-rank framework for multi-step ahead time-series forecasting which aims to optimise the relative ordering of the forecast. A local learning technique is incorporated in the framework to tackle the approximate inference of the time-series value given the forecast of its relative orderings at each time epoch along the timeline. A dynamic prediction scheme is proposed to integrate the proposed learning-to-rank model and the local learning in an iterative manner. A combined framework results in an improvement in terms of the CI, which is robust to subtle changes of the observed values. By comparing the proposed framework with a series of conventional methods on financial time series across different types of asset classes, we empirically verified that the proposed framework outperforms its competing methods through the lens of the CI. A comprehensive examinations of the proposed method is provided from multiple perspectives, e.g., the impact of the learning-to-rank model in time-series forecasting, the use of the proposed method, for different categories of time series and under different horizons of interest.

Chapter 6

Conclusion and Future Works

6.1 Conclusion

In this dissertation, the data variety and veracity issues in predicting with structured data have been studied by raising three fundamental questions:

1. **DOES** atomic-level similarity provide constructive information for predictive modeling in materials informatics?

2. **WHAT** is the proper data representation to mitigate the conflicting observations, given unconcordant pairwise relationships? **HOW** to improve the predictive performance?

3. **IS** concordance an alternative and meaningful optimization objective for time-series forecasting? **HOW** to leverage it as a complement to conventional tracking errors?

We developed three types of approaches that answer the above questions accordingly.

1. Atomic-level similarity can provide constructive information for predictive modeling in materials informatics. By using fine-grained graph kernel and domain knowledge, e.g. the adjacency of atomic elements in the periodic table, the predictive model can exhibit superior prediction power by using only a small amount of labeled data.

2. Structured representation of players that separately expresses the strength and the weakness of a player can mitigate the negative impact of conflicting observations or unconcordant pairwise relationships. In order to improve the expressiveness or predictive performance, developing deeper numerical interaction between the embedded representations via transitive matrices or structured deep neural networks can help.

3. Ranking of temporal observations is an alternative and meaningful optimization objective for time-series forecasting. The relative ordering of temporal observations can be optimized by learning to rank techniques and this optimization is robust to subtle changes in observed value, which mitigates the challenge of data veracity for time series.

For the first approach, a novel predictive model for materials informatics is proposed based on random walk graph kernel. The model utilizes atomic-level and bond-level similarities of the graph elements in a materials compound. The similarity is estimated based on one of the most fundamental domain knowledge in materials science and in chemistry, i.e., the periodic table of the elements, because the element-level adjacency in the periodic table is a good descriptor of electromagnetic properties. By constructing fine-grained kernels for such domain knowledge, additional intelligence from the domain experts can be incorporated into the predictive analytics framework. To reduce the complexity of the computation, a minimum spanning tree algorithm can be used to cut off the redundant chemical bonds and carve out the critical vertices and edges in a graph representation. As a result, the computational complexity of a kernel update reduces from $O(|V|^3)$ to $O(|V|^2)$. Experimental results on real-world materials informatics database show that, in comparison to the methods that rely on conventional handicraft features, the predictive performance of the proposed method is significant better, especially when the availability of labeled sample is limited.

For the second approach, two novel techniques are proposed to improve the expressiveness of intransitivity by incorporating additional numerical interaction between model parameters. The first technique is a novel multi-dimensional representation of objects where a unified structured representation of objects is proposed to handle various scoring attribution, including the interaction between objects and the intrinsic strength of each individual. Regarding the expressiveness of the proposed method, a simple and constructive example is illustrated to show how the proposed model generalises its related works. On the parameter estimation, a stochastic gradient algorithm based on alternating direction methods of multipliers is devised. Through missing value inference experiments on recommender system, food preference and online gaming platform, we demonstrate that the proposed method improves the prediction accuracy. Moreover, we provide an extensive investigation of the universal existence of intransitive relationships between objects in real-world datasets. This highlights the importance of dealing with intransitivity in ranking problems. The second approach we proposed is a novel end-to-end intransitivity modeling technique. Different from the techniques presented previously, a structured multi-layer neural network is used for the parameter estimation instead of the legacy logistic function. Through experiments on synthetic data with a controlled rank, we show that the proposed end-to-end deep neural network approach is superior to the shallow models. This opens the gateway to the development of more expressive network structures. Through missing value inference experiments, we demonstrate that the proposed method is a simpler yet more effective method to improve the prediction accuracy for the modeling of pairwise comparison and matchup.

For the third approach, a novel time-series forecasting method is proposed to optimize and infer the relative relationship between temporal observations. The model is robust to noisy temporal observations by its discrete nature and directly optimizes ranking objectives. To make the forecast comparable to what is predicted by the legacy tracking-error-focused techniques, we develop a local learning technique based on nearest neighbor model. The predictive performance can be further improved by implementing an overlaying dynamic prediction strategy that warps up the heterogeneous modules. The proposed method significantly improves the prediction of relative relationship between temporal observations. Extensive experiments on financial data concludes the practical concerns for different types of time-series and in various lengths of forward-looking horizon.

6.2 Future Works

As for future works, two directions for predicting with structured data are noteworthy.

Firstly, it is recognized that the variety and veracity of structured data will inevitably become more challenging than what we can foresee at present, because they are the forefront business-related aspects when we handle big data. As with the wide adoption of deep neural networks, learning algorithms that handle structured patterns should be more explainable and robust against noises and adversarial samples. Going forward, representation learning techniques which envision the automatic discovery of patterns from data worth further investigation. Besides, techniques that can integrate the innovative techniques and the legacy techniques and make them complement with each other will guide us towards an all-round status in predictive modeling. A primitive effort in this line of research is studied in Chapter 5 of this dissertation and we look forward to future research efforts in this direction.

Secondly, regarding the underlying techniques in predicting with structured data, e.g., graphs, rank, and time-series, learning with finer-grained structured representation of plain objects remains as a green field. The technical challenge behind is the algorithmic efficiency, because the complexity of the algorithm may grow exponentially when fine-grained patterns are investigated. Therefore, Revisiting the foundation of algorithms, from graph search algorithms that can systematically enumerate the candidate structures to dynamic programming that is the best practice in deterministic inference, will become the lancet for the next generation of learning algorithms when predicting with structured data.

List of Publications

1. **Jiuding Duan**, Hisashi Kashima. Learning to Rank for Multi-Step Ahead Time-Series Forecasting. *IEEE Access* 9 (2021): 49372-49386, 2021.

2. Yan Gu, **Jiuding Duan**, and Hisashi Kashima. An Intransitivity Model for Matchup and Pairwise Comparison. *In Proceedings of the 25th International Conference on Pattern Recognition (ICPR)*, pp. 692-698, IEEE, 2020.

3. **Jiuding Duan**, Jiyi Li, Yukino Baba, and Hisashi Kashima. A generalized model for multidimensional intransitivity. *In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pp. 840-852. Springer, Cham, 2017.

4. **Jiuding Duan**, Atsuto Seko, and Hisashi Kashima. "Quantum Energy Prediction Using Graph Kernel." *In Proceedings of 2015 IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC)*, pp. 1651-1656. IEEE, 2015.

Bibliography

[1] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

[2] Vasant Dhar. Data science and prediction. *Communications of the ACM*, 56(12):64–73, 2013.

[3] Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J McLachlan, Angus Ng, Bing Liu, S Yu Philip, et al. Top 10 algorithms in data mining. *Knowledge and information systems*, 14(1):1–37, 2008.

[4] Rampi Ramprasad, Rohit Batra, Ghanshyam Pilania, Arun Mannodi-Kanakkithodi, and Chiho Kim. Machine learning in materials informatics: recent applications and prospects. *npj Computational Materials*, 3(1):1–13, 2017.

[5] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.

[6] X Gonze, J-M Beuken, R Caracas, F Detraux, M Fuchs, G-M Rignanese, L Sindic, M Verstraete, G Zerah, F Jollet, et al. First-principles computation of material properties: the ABINIT software project. *Computational Materials Science*, 25(3):478–492, 2002.

[7] Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, and O Anatole von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. *Physical Review Letters*, 108(5):058301, 2012.

[8] Grégoire Montavon, Katja Hansen, Siamac Fazli, Matthias Rupp, Franziska Biegler, Andreas Ziehe, Alexandre Tkatchenko, Anatole V Lilienfeld, and Klaus-Robert Müller. Learning invariant representations of molecules for atomization energy prediction. In *Advances in Neural Information Processing Systems*, pages 440–448, 2012.

[9] KT Schütt, H Glawe, F Brockherde, A Sanna, KR Müller, and EKU Gross. How to represent crystal structures for machine learning: Towards fast prediction of electronic properties. *Physical Review B*, 89(20):205118, 2014.

[10] Charles Kittel, Paul McEuen, and Paul McEuen. *Introduction to Solid State Physics*, volume 8. Wiley New York, 1976.

[11] Grégoire Montavon, Matthias Rupp, Vivekanand Gobre, Alvaro Vazquez-Mayagoitia, Katja Hansen, Alexandre Tkatchenko, Klaus-Robert Müller, and O Anatole von Lilienfeld. Machine learning of molecular electronic properties in chemical compound space. *New Journal of Physics*, 15(9):095003, 2013.

[12] T Hastie, J Friedman, and R Tibshirani. *The Elements of Statistical Learning*, volume 2. Springer, 2009.

[13] Hisashi Kashima, Koji Tsuda, and Akihiro Inokuchi. Marginalized kernels between labeled graphs. In *Proceedings of the 20th ICML*, volume 3, pages 321–328, 2003.

[14] Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-Lehman graph kernels. *Journal of Machine Learning Research*, 12:2539–2561, 2011.

[15] Karsten M Borgwardt and H-P Kriegel. Shortest-path kernels on graphs. In *Proceedings of 5th ICDM*, pages 8–16. IEEE, 2005.

[16] Risi Kondor, Nino Shervashidze, and Karsten M Borgwardt. The graphlet spectrum. In *Proceedings of the 26th ICML*, pages 529–536. ACM, 2009.

[17] Atsuto Seko, Kazuki Shitara, and Isao Tanaka. Efficient determination of alloy ground-state structures. *Physical Review B*, 90(17):174104, 2014.

[18] Materialproject: http://materialproject.org.

[19] ICSD database: https://icsd.fiz-karlsruhe.de/.

[20] Hiroto Saigo, Masahiro Hattori, Hisashi Kashima, and Koji Tsuda. Reaction graph kernels predict EC numbers of unknown enzymatic reactions in plant secondary metabolism. *BMC Bioinformatics*, 11(Suppl 1):S31, 2010.

[21] Marc G Genton. Classes of kernels for machine learning: a statistics perspective. *Journal of Machine Learning Research*, 2:299–312, 2002.

[22] Hiroshi Kajino, Yuta Tsuboi, and Hisashi Kashima. A convex formulation for learning from crowds. In *Proceedings of AAAI*, 2012.

[23] Andriy Mnih and Geoffrey E Hinton. A scalable hierarchical distributed language model. In *Advances in Neural Information Processing Systems*, pages 1081–1088, 2009.

[24] David Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

[25] Hiroshi Kajino, Yuta Tsuboi, and Hisashi Kashima. Clustering crowds. In *Proceedings of AAAI*, 2013.

[26] Matthew Fisher, Manolis Savva, and Pat Hanrahan. Characterizing structural relationships in scenes using graph kernels. In *ACM Transactions on Graphics*, volume 30, page 34. ACM, 2011.

[27] Hisashi Kashima, Koji Tsuda, and Akihiro Inokuchi. Kernels for graphs. *Kernel Methods in Computational Biology*, 39(1):101–113, 2004.

[28] Atsuto Seko, Akira Takahashi, and Isao Tanaka. Sparse representation for a potential energy surface. *Physical Review B*, 90(2):024101, 2014.

[29] Zaïd Harchaoui and Francis Bach. Image classification with segmentation graph kernels. In *Proceedings of the 20th CVPR*, pages 1–8. IEEE, 2007.

[30] Artem R Oganov. *Modern Methods of Crystal Structure Prediction*. John Wiley & Sons, 2011.

[31] AH Castro Neto, F Guinea, NMR Peres, Kostya S Novoselov, and Andre K Geim. The electronic properties of graphene. *Reviews of modern physics*, 81(1):109, 2009.

[32] Candace K Chan, Hailin Peng, Gao Liu, Kevin McIlwrath, Xiao Feng Zhang, Robert A Huggins, and Yi Cui. High-performance lithium battery anodes using silicon nanowires. *Nature nanotechnology*, 3(1):31–35, 2008.

[33] Xiao-Ming Wu, Zhenguo Li, Anthony M So, John Wright, and Shih-Fu Chang. Learning with partially absorbing random walks. In *Advances in Neural Information Processing Systems*, pages 3077–3085, 2012.

[34] Kevin P Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[35] Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, et al. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL materials*, 1(1):011002, 2013.

[36] Cynthia Dwork, Ravi Kumar, Moni Naor, and Dandapani Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th international conference on World Wide Web*, pages 613–622, 2001.

[37] Yu-Ting Liu, Tie-Yan Liu, Tao Qin, Zhi-Ming Ma, and Hang Li. Supervised rank aggregation. In *Proceedings of the 16th international conference on World Wide Web*, pages 481–490, 2007.

[38] Li-Xin Wang and Feng Wan. Structured neural networks for constrained model predictive control. *Automatica*, 37(8):1235–1243, 2001.

[39] Gediminas Adomavicius, Ramesh Sankaranarayanan, Shahana Sen, and Alexander Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems (TOIS)*, 23(1):103–145, 2005.

[40] Linas Baltrunas and Francesco Ricci. Context-based splitting of item ratings in collaborative filtering. In *Proceedings of the third ACM conference on Recommender systems*, pages 245–248, 2009.

[41] Steffen Rendle, Zeno Gantner, Christoph Freudenthaler, and Lars Schmidt-Thieme. Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 635–644, 2011.

[42] Joel Huber, John W Payne, and Christopher Puto. Adding asymmetrically dominated alternatives: Violations of regularity and the similarity hypothesis. *Journal of consumer research*, 9(1):90–98, 1982.

[43] Itamar Simonson and Amos Tversky. Choice in context: Tradeoff contrast and extremeness aversion. *Journal of marketing research*, 29(3):281–295, 1992.

[44] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[45] Lei Zhang, Jun Wu, Zhong-Cun Wang, and Chong-Jun Wang. A factor-based model for context-sensitive skill rating systems. In *2010 22nd IEEE International Conference on Tools with Artificial Intelligence*, volume 2, pages 249–255. IEEE, 2010.

[46] Ryan Prescott Adams, George E Dahl, and Iain Murray. Incorporating side information in probabilistic matrix factorization with Gaussian processes. *arXiv preprint arXiv:1003.4944*, 2010.

[47] Prem Gopalan, Jake M Hofman, and David M Blei. Scalable recommendation with poisson factorization. *arXiv preprint arXiv:1311.1704*, 2013.

[48] Ding Zhou, Shenghuo Zhu, Kai Yu, Xiaodan Song, Belle L Tseng, Hongyuan Zha, and C Lee Giles. Learning multiple graphs for document recommendations. In *Proceedings of the 17th international conference on World Wide Web*, pages 141–150, 2008.

[49] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136, 2007.

[50] James P Keener. The Perron–Frobenius theorem and the ranking of football teams. *SIAM review*, 35(1):80–93, 1993.

[51] David R Hunter et al. Mm algorithms for generalized Bradley-Terry models. *The annals of statistics*, 32(1):384–406, 2004.

[52] David Causeur and François Husson. A 2-dimensional extension of the Bradley-Terry model for paired comparisons. *Journal of Statistical Planning and Inference*, 135(2):245–259, 2005.

[53] Roger R Davidson. On extending the Bradley-Terry model to accommodate ties in paired comparison experiments. *Journal of the American Statistical Association*, 65(329):317–328, 1970.

[54] Pedro Linares. Are inconsistent decisions better? an experiment with pairwise comparisons. *European Journal of Operational Research*, 193(2):492–498, 2009.

[55] Tzu-Kuo Huang, Chih-Jen Lin, and Ruby C Weng. Ranking individuals by group comparisons. *Journal of Machine Learning Research*, 9(Oct):2187–2216, 2008.

[56] Ralf Herbrich, Tom Minka, and Thore Graepel. Trueskill™: A bayesian skill rating system. In *Advances in Neural Information Processing Systems 19*, pages 569–576. MIT Press, 2007.

[57] Pierre Dangauthier, Ralf Herbrich, Tom Minka, and Thore Graepel. TrueSkill through time: Revisiting the history of chess. In *Advances in neural information processing systems*, pages 337–344, 2008.

[58] Mark E Glickman. A comprehensive guide to chess ratings. *American Chess Journal*, 3(1):59–102, 1995.

[59] David Causeur and François Husson. A 2-dimensional extension of the Bradley–Terry model for paired comparisons. *Journal of statistical planning and inference*, 135(2):245–259, 2005.

[60] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *ACM SIGIR Forum*, volume 51, pages 227–234. ACM New York, NY, USA, 2017.

[61] Michel Regenwetter, Jason Dana, and Clintin P Davis-Stober. Transitivity of preferences. *Psychological review*, 118(1):42, 2011.

[62] Manuela Cattelan. Models for paired comparison data: A review with emphasis on dependent data. *Statistical Science*, pages 412–433, 2012.

[63] Mohsen Jamali and Martin Ester. A transitivity aware matrix factorization model for recommendation in social networks. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[64] David F Gleich and Lek-heng Lim. Rank aggregation via nuclear norm minimization. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 60–68, 2011.

[65] Louis L Thurstone. A law of comparative judgment. *Psychological review*, 34(4):273, 1927.

[66] Joshua E Menke and Tony R Martinez. A Bradley–Terry artificial neural network model for individual ratings in group competitions. *Neural computing and Applications*, 17(2):175–186, 2008.

[67] Thore Graepel, Tom Minka, and R TrueSkill Herbrich. A Bayesian skill rating system. *Advances in Neural Information Processing Systems*, 19:569–576, 2007.

[68] Manuela Cattelan, Cristiano Varin, and David Firth. Dynamic Bradley–Terry modelling of sports tournaments. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 62(1):135–150, 2013.

[69] Nicolaus Tideman. *Collective decisions and voting: the potential for public choice*. Routledge, 2017.

[70] Toshihiro Kamishima. Nantonac collaborative filtering: recommendation based on order responses. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 583–588, 2003.

[71] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[72] Jiuding Duan, Jiyi Li, Yukino Baba, and Hisashi Kashima. A generalized model for multidimensional intransitivity. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 840–852. Springer, 2017.

[73] Shuo Chen and Thorsten Joachims. Predicting matchups and preferences in context. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 775–784, 2016.

[74] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[75] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[76] R Duncan Luce. *Individual choice behavior: A theoretical analysis*. Courier Corporation, 2012.

[77] Shuo Chen and Thorsten Joachims. Modeling intransitivity in matchup and comparison data. In *Proceedings of the ninth acm international conference on web search and data mining*, pages 227–236. ACM, 2016.

[78] Donald B Johnson. Finding all the elementary circuits of a directed graph. *SIAM Journal on Computing*, 4(1):77–84, 1975.

[79] Amir Globerson, Gal Chechik, Fernando Pereira, and Naftali Tishby. Euclidean embedding of co-occurrence data. *Journal of Machine Learning Research*, 8(Oct):2265–2295, 2007.

[80] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd Annual*

International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 230–237. ACM, 1999.

[81] Nicolaus Tideman. *Collective decisions and voting: the potential for public choice*. Ashgate Publishing, Ltd., 2006.

[82] Linxia Gong, Xiaochuan Feng, Dezhi Ye, Hao Li, Runze Wu, Jianrong Tao, Changjie Fan, and Peng Cui. Optmatch: Optimized matchmaking via modeling the high-order interactions on the arena. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2300–2310, 2020.

[83] Rahul Makhijani and Johan Ugander. Parametric models for intransitivity in pairwise rankings. In *The World Wide Web Conference*, pages 3056–3062, 2019.

[84] Lucas Maystre, Victor Kristof, and Matthias Grossglauser. Pairwise comparisons with flexible time-dynamics. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1236–1246, 2019.

[85] Arjun Seshadri, Alex Peysakhovich, and Johan Ugander. Discovering context effects from raw choice data. In *International Conference on Machine Learning*, pages 5660–5669, 2019.

[86] Yao Li, Minhao Cheng, Kevin Fujii, Fushing Hsieh, and Cho-Jui Hsieh. Learning from group comparisons: exploiting higher order interactions. In *Advances in Neural Information Processing Systems*, pages 4981–4990, 2018.

[87] Bryan Lim, Stefan Zohren, and Stephen Roberts. Enhancing time-series momentum strategies using deep neural networks. *The Journal of Financial Data Science*, 1(4):19–38, 2019.

[88] Zihao Zhang, Stefan Zohren, and Stephen Roberts. Deep reinforcement learning for trading. *The Journal of Financial Data Science*, 2(2):25–40, 2020.

[89] Bryan Lim and Stefan Zohren. Time series forecasting with deep learning: A survey. *arXiv preprint arXiv:2004.13408*, 2020.

[90] Trent Spears, Stefan Zohren, and Stephen Roberts. Investment sizing with deep learning prediction uncertainties for high-frequency eurodollar futures trading. *Available at SSRN 3664497*, 2020.

[91] Tobias J Moskowitz, Yao Hua Ooi, and Lasse Heje Pedersen. Time series momentum. *Journal of financial economics*, 104(2):228–250, 2012.

[92] LE Vincent and Nicolas Thome. Shape and time distortion loss for training deep time series forecasting models. In *Advances in Neural Information Processing Systems*, pages 4189–4201, 2019.

[93] Jui-Sheng Chou, Dinh-Nhat Truong, and Thuy-Linh Le. Interval forecasting of financial time series by accelerated particle swarm-optimized multi-output machine learning system. *IEEE Access*, 8:14798–14808, 2020.

[94] Xiurui Hou, Kai Wang, Jie Zhang, and Zhi Wei. An enriched time-series forecasting framework for long-short portfolio strategy. *IEEE Access*, 8:31992–32002, 2020.

[95] Shihao Gu, Bryan Kelly, and Dacheng Xiu. Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5):2223–2273, 2020.

[96] Sendhil Mullainathan and Jann Spiess. Machine learning: an applied econometric approach. *Journal of Economic Perspectives*, 31(2):87–106, 2017.

[97] Shihao Gu, Bryan Kelly, and Dacheng Xiu. Autoencoder asset pricing models. *Journal of Econometrics*, 2020.

[98] Philipp Krueger, Zacharias Sautner, and Laura T Starks. The importance of climate risks for institutional investors. *The Review of Financial Studies*, 33(3):1067–1111, 2020.

[99] Hirotugu Akaike. Fitting autoregressive models for prediction. *Annals of the institute of Statistical Mathematics*, 21(1):243–247, 1969.

[100] George EP Box and David A Pierce. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American statistical Association*, 65(332):1509–1526, 1970.

[101] Javier Contreras, Rosario Espinola, Francisco J Nogales, and Antonio J Conejo. Arima models to predict next-day electricity prices. *IEEE transactions on power systems*, 18(3):1014–1020, 2003.

[102] G Peter Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.

[103] Shogo Hayashi, Akira Tanimoto, and Hisashi Kashima. Long-term prediction of small time-series data using generalized distillation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[104] Eugene F Fama and Kenneth R French. Multifactor explanations of asset pricing anomalies. *The journal of finance*, 51(1):55–84, 1996.

[105] Shihao Gu, Bryan Kelly, and Dacheng Xiu. Empirical asset pricing via machine learning. Technical report, National Bureau of Economic Research, 2018.

[106] Eugene F Fama. Random walks in stock market prices. *Financial analysts journal*, 51(1):75–80, 1995.

[107] Ameeth Kanawaday and Aditya Sane. Machine learning for predictive maintenance of industrial machines using iot sensor data. In *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pages 87–90. IEEE, 2017.

[108] Kenji Yamanishi and Jun-ichi Takeuchi. A unifying framework for detecting outliers and change points from non-stationary time series data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 676–681, 2002.

[109] Ralph SJ Koijen and Stijn Van Nieuwerburgh. Predictability of returns and cash flows. *Annu. Rev. Financ. Econ.*, 3(1):467–491, 2011.

[110] James D Hamilton. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica: Journal of the Econometric Society*, pages 357–384, 1989.

[111] Ben S Bernanke and Jean Boivin. Monetary policy in a data-rich environment. *Journal of Monetary Economics*, 50(3):525–546, 2003.

[112] Yemane Wolde-Rufael. Electricity consumption and economic growth: a time series experience for 17 african countries. *Energy policy*, 34(10):1106–1114, 2006.

[113] Gary Koop and Dimitris Korobilis. *Bayesian multivariate time series methods for empirical macroeconomics*. Now Publishers Inc, 2010.

[114] Marco Cuturi and Mathieu Blondel. Soft-dtw: a differentiable loss function for time-series. In *International Conference on Machine Learning*, pages 894–903, 2017.

[115] Tao Xiong, Yukun Bao, and Zhongyi Hu. Beyond one-step-ahead forecasting: evaluation of alternative multi-step-ahead forecasting models for crude oil prices. *Energy Economics*, 40:405–415, 2013.

[116] Alexander G Parlos, Omar T Rais, and Amir F Atiya. Multi-step-ahead prediction using dynamic recurrent neural networks. *Neural networks*, 13(7):765–786, 2000.

[117] Xia Hong and SA Billings. Time series multistep-ahead predictability estimation and ranking. *Journal of Forecasting*, 18(2):139–149, 1999.

[118] Yukun Bao, Tao Xiong, and Zhongyi Hu. Multi-step-ahead time series prediction using multiple-output support vector regression. *Neurocomputing*, 129:482–493, 2014.

[119] Souhaib Ben Taieb, Gianluca Bontempi, Amir F Atiya, and Antti Sorjamaa. A review and comparison of strategies for multi-step ahead time series forecasting based on the nn5 forecasting competition. *Expert systems with applications*, 39(8):7067–7083, 2012.

[120] Gianluca Bontempi, Souhaib Ben Taieb, and Yann-Aël Le Borgne. Machine learning strategies for time series forecasting. In *European business intelligence summer school*, pages 62–77. Springer, 2012.

[121] Ibrahim Moghram and Saifur Rahman. Analysis and evaluation of five short-term load forecasting techniques. *IEEE Transactions on power systems*, 4(4):1484–1491, 1989.

[122] Vincent Le Guen and Nicolas Thome. Probabilistic time series forecasting with shape and temporal diversity. *Advances in Neural Information Processing Systems*, 33, 2020.

[123] Mithat Gönen and Glenn Heller. Concordance probability and discriminatory power in proportional hazards regression. *Biometrika*, 92(4):965–970, 2005.

[124] Tarun Chordia and Lakshmanan Shivakumar. Momentum, business cycle, and time-varying expected returns. *The Journal of Finance*, 57(2):985–1019, 2002.

[125] Tom Minka, Ryan Cleven, and Yordan Zaykov. Trueskill 2: An improved bayesian skill rating system. *Tech. Rep.*, 2018.

[126] Hai-Tao Yu, Adam Jatowt, Hideo Joho, Joemon M Jose, Xiao Yang, and Long Chen. Wassrank: Listwise document ranking using optimal transport theory. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 24–32, 2019.

[127] Christiane Lemke and Bogdan Gabrys. Meta-learning for time series forecasting and forecast combination. *Neurocomputing*, 73(10-12):2006–2016, 2010.

[128] Nick Baltas and Robert Kosowski. Demystifying time-series momentum strategies: Volatility estimators, trading rules and pairwise correlations. *Market Momentum: Theory and Practice", Wiley*, 2020.

[129] Jamil Baz, Nicolas Granger, Campbell R Harvey, Nicolas Le Roux, and Sandy Rattray. Dissecting investment strategies in the cross section and time series. *Available at SSRN 2695101*, 2015.

[130] Thorsten Joachims. A support vector method for multivariate performance measures. In *Proceedings of the 22nd international conference on Machine learning*, pages 377–384, 2005.

[131] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[132] Hong Zeng and Yiu-ming Cheung. Feature selection and kernel learning for local learning-based clustering. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1532–1547, 2010.

[133] Xingyi Cheng, Ruiqing Zhang, Jie Zhou, and Wei Xu. Deeptransport: Learning spatial-temporal dependency for traffic condition forecasting. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

[134] Tao Lin, Tian Guo, and Karl Aberer. Hybrid neural networks for learning the trend in time series. In *Proceedings of the twenty-sixth international joint conference on artificial intelligence*, pages 2273–2279, 2017.

[135] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Departmental Papers, University of Pennsylvania*, 2001.

[136] Vassilis Assimakopoulos and Konstantinos Nikolopoulos. The theta model: a decomposition approach to forecasting. *International journal of forecasting*, 16(4):521–530, 2000.

[137] Evangelos Spiliotis, Vassilios Assimakopoulos, and Spyros Makridakis. Generalizing the theta method for automatic forecasting. *European Journal of Operational Research*, 284(2):550–558, 2020.

[138] Benjamin Kedem and Konstantinos Fokianos. *Regression models for time series analysis*, volume 488. John Wiley & Sons, 2005.

[139] Kung-Yee Liang and Scott L Zeger. A class of logistic regression models for multivariate binary time series. *Journal of the American Statistical Association*, 84(406):447–451, 1989.

[140] Hanwei Wu, Ather Gattami, and Markus Flierl. Conditional mutual information-based contrastive loss for financial time series forecasting. *arXiv preprint arXiv:2002.07638*, 2020.

[141] Harald Steck, Balaji Krishnapuram, Cary Dehing-Oberije, Philippe Lambin, and Vikas C Raykar. On ranking in survival analysis: Bounds on the concordance index. In *Advances in neural information processing systems*, pages 1209–1216, 2008.

[142] Paul Blanche, Michael W Kattan, and Thomas A Gerds. The c-index is not proper for the evaluation of-year predicted risks. *Biostatistics*, 20(2):347–357, 2019.

[143] Marisa Mohr, Florian Wilhelm, Mattis Hartwig, Ralf Möller, and Karsten Keller. New approaches in ordinal pattern representations for multivariate time series. In *FLAIRS Conference*, pages 124–129, 2020.

[144] Enrico Longato, Martina Vettoretti, and Barbara Di Camillo. A practical perspective on the concordance index for the evaluation and selection of prognostic time-to-event models. *Journal of Biomedical Informatics*, page 103496, 2020.

[145] Ulf Herold and Raimond Maurer. Structural positions and risk budgeting: Quantifying the impact of structural positions and deriving implications for active portfolio management. *Journal of Asset Management*, 9(2):149–157, 2008.

[146] Gang Liu, Fuyuan Xiao, Chin-Teng Lin, and Zehong Cao. A fuzzy interval time-series energy and financial forecasting model using network-based multiple time-frequency spaces and the induced-ordered weighted averaging aggregation operation. *IEEE Transactions on Fuzzy Systems*, 28(11):2677–2690, 2020.

[147] Shengzhong Mao and Fuyuan Xiao. A novel method for forecasting construction cost index based on complex network. *Physica A: Statistical Mechanics and its Applications*, 527:121306, 2019.

[148] Gang Liu and Fuyuan Xiao. Time series data fusion based on evidence theory and owa operator. *Sensors*, 19(5):1171, 2019.

[149] Rangasami L Kashyap. Optimal choice of ar and ma parts in autoregressive moving average models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2):99–104, 1982.

[150] Hui-Kuang Yu. Weighted fuzzy time series models for taiex forecasting. *Physica A: Statistical Mechanics and its Applications*, 349(3-4):609–624, 2005.

[151] Ronald R Yager. Generalized dempster–shafer structures. *IEEE Transactions on Fuzzy Systems*, 27(3):428–435, 2018.

[152] Souhaib Ben Taieb and Amir F Atiya. A bias and variance analysis for multistep-ahead time series forecasting. *IEEE transactions on neural networks and learning systems*, 27(1):62–76, 2015.

[153] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

[154] Wei Bao, Jun Yue, and Yulei Rao. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one*, 12(7):e0180944, 2017.

[155] Rob J Hyndman, Roman A Ahmed, George Athanasopoulos, and Han Lin Shang. Optimal combination forecasts for hierarchical time series. *Computational statistics & data analysis*, 55(9):2579–2589, 2011.

[156] Lucas Lacasa, Bartolo Luque, Fernando Ballesteros, Jordi Luque, and Juan Carlos Nuno. From time series to complex networks: The visibility graph. *Proceedings of the National Academy of Sciences*, 105(13):4972–4975, 2008.

[157] Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. *Advances in neural information processing systems*, 31:7785–7794, 2018.

[158] Hristos Tyralis and Georgia Papacharalampous. Variable selection in time series forecasting using random forests. *Algorithms*, 10(4):114, 2017.

[159] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30:3146–3154, 2017.

[160] Gary Koop, Dimitris Korobilis, and Davide Pettenuzzo. Bayesian compressed vector autoregressions. *Journal of Econometrics*, 210(1):135–154, 2019.

[161] Jinxing Che and Jianzhou Wang. Short-term electricity prices forecasting based on support vector regression and auto-regressive integrated moving average modeling. *Energy Conversion and Management*, 51(10):1911–1917, 2010.

[162] Philippe Goulet Coulombe, Maxime Leroux, Dalibor Stevanovic, and Stéphane Surprenant. How is machine learning useful for macroeconomic forecasting? *arXiv preprint arXiv:2008.12477*, 2020.

[163] Nattapol Aunsri and Paponpat Taveeapiradeecharoen. A time-varying bayesian compressed vector autoregression for macroeconomic forecasting. *IEEE Access*, 8:192777–192786, 2020.

[164] Aaron Smalter Hall et al. Machine learning approaches to macroeconomic forecasting. *The Federal Reserve Bank of Kansas City Economic Review*, 103(63):2, 2018.

[165] Kohei Maehashi and Mototsugu Shintani. Macroeconomic forecasting using factor models and machine learning: an application to japan. *Journal of the Japanese and International Economies*, 58:101104, 2020.

[166] Jin-Kyu Jung, Manasa Patnam, and Anna Ter-Martirosyan. *An algorithmic crystal ball: forecasts-based on machine learning*. International Monetary Fund, 2018.

[167] Marlon Nunez, Raul Fidalgo-Merino, and Rafael Morales. An event-based predictive modelling approach: An application in macroeconomics. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1169–1174. IEEE, 2018.

[168] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *the Journal of machine Learning research*, 9:1871–1874, 2008.

[169] Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012.

[170] Andreas Mayr and Matthias Schmid. Boosting the concordance index for survival data–a unified framework to derive and evaluate biomarker combinations. *PloS one*, 9(1):e84483, 2014.

[171] Sebastian Pölsterl. scikit-survival: A library for time-to-event analysis built on top of scikit-learn. *Journal of Machine Learning Research*, 21(212):1–6, 2020.

[172] Eric JG Sijbrands, Erik Tornij, and Sietske J Homsma. Mortality risk prediction by an insurance company and long-term follow-up of 62,000 men. *PloS one*, 4(5):e5457, 2009.

[173] T. Joachims. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226, 2006.

[174] Markus Löning, Anthony Bagnall, Sajaysurya Ganesh, Viktor Kazakov, Jason Lines, and Franz J Király. sktime: A unified interface for machine learning with time series. *arXiv preprint arXiv:1909.07872*, 2019.

[175] Alexander Alexandrov, Konstantinos Benidis, Michael Bohlke-Schneider, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Danielle C Maddix, Syama Rangapuram, David Salinas, Jasper Schulz, et al. Gluonts: Probabilistic and neural time series modeling in python. *Journal of Machine Learning Research*, 21(116):1–6, 2020.